

# 100 DICAS PARA MSX



**N**Editora  
**Aleph**

**TÉCNICAS E  
TRUQUES DE  
PROGRAMAÇÃO**

**RENATO DA SILVA OLIVEIRA**

# **100 DICAS**

## **PARA**

# **MSX**

### **TÉCNICAS E TRUQUES DE PROGRAMAÇÃO**

**AUTORES:**

Henrique de Figueredo Luz  
Luís Tarcísio de Carvalho Jr.  
Milton Maldonado Jr. (The Pilot)  
Pierluigi Piazzi  
Renato da Silva Oliveira  
Rubens Pereira Silva Jr.

**3ª EDIÇÃO  
1989**



**EXPEDIENTE**

**Coordenação Editorial:** PIERLUIGI PIAZZI  
**Coordenação Didática:** BETTY FROMER PIAZZI  
**Produção Editorial:** ROSA KOGAN FROMER  
**Editoração:** RENATO DA SILVA OLIVEIRA  
**Ilustrações:** DURVALY ODILON NICOLETTI



**ALEPH Publicações e**  
**Assessoria Pedagógica Ltda**  
R. Dr. Luiz Migliano 1110 cj.301  
05711 São Paulo SP  
Caixa Postal 20707 CEP 01498  
Tel: (011) 843-3202

**Dados de Catalogação na Publicação (CIP) Internacional**  
**(Câmara Brasileira do Livro, SP, Brasil)**

O51c Oliveira, Renato da Silva, 1960-  
100 dicas para MSX : técnicas e truques de programação / Renato da Silva Oliveira. — 1. ed. — São Paulo : Aleph, 1988.  
(Coleção MSX)

1. MSX (Computadores) - Programação I. Título.  
II. Série.

88-0118

CDD-001.642

**Índices para catálogo sistemático:**

1. MSX : Computadores : Programação : Processamento de dados 001.642

# CEM DICAS PARA O MSX

## SUMARIO

APRESENTACAO .....	6
1 - DICAS SOBRE O TECLADO .....	7
2 - DICAS PARA USAR O VIDEO .....	21
3 - DICAS SONORAS .....	88
4 - DICAS PARA CASSETTE .....	105
5 - DICAS PARA IMPRESSORA .....	117
6 - DICAS PARA O DRIVE .....	134
7 - DICAS DE PROCESSAMENTO .....	153
8 - NOTAS SOBRE A BIBLIOGRAFIA RECOMENDADA ..	191

## APRESENTAÇÃO

A grande maioria dos usuários de microcomputadores MSX nunca teve outro micro antes. Apesar de terem entrado no mundo da computação pela porta da frente, é provável que esses usuários tenham alguma dificuldade inicial para programar suas máquinas. Poucos deles têm, de imediato, uma visão muito clara dos poderosos recursos de que dispõem.

Este livro contém mais de cem dicas de programação já prontas para serem usadas. Elas permitem um aproveitamento muito maior dos recursos dos MSX mesmo pelos usuários iniciantes e, certamente, acrescentam conhecimentos valiosos aos programadores mais experientes.

Apesar de serem geralmente independentes umas das outras, as dicas estão agrupadas em 7 capítulos, de acordo com a função a que elas se destinam.

O leitor poderá usá-las em outros programas para otimizá-los ou obter resultados específicos.

Cada dica está apresentada de forma bem prática e resumida de modo a tornar seu uso imediato muito fácil. As explicações nem sempre são detalhadas mas, sempre que possível, faz-se referência a textos de outros livros onde o assunto é comentado mais extensamente.

Apesar de muitas dicas serem programas em Linguagem de Máquina, optamos pelo uso exclusivo da linguagem BASIC para gerá-las. Isso torna a digitação mais fácil e o uso mais imediato. Todos os programas estão em BASIC, mas mesmo assim, para diminuir a ocorrência de erros de digitação, junto a cada listagem pode-se encontrar em "vídeo inverso" sua SOMA SINTÁTICA. As explicações sobre a produção e uso dessa "soma" são dadas na dica 7.1 (página 178) e ela é, portanto, a primeira dica que deve ser lida.

Lembre-se que a SOMA TOTAL de um programa, mesmo coincidindo com a apresentada no livro, não elimina totalmente a ocorrência de erros de digitação, mas apenas a reduz. Deve-se também considerar que a soma que apresentamos foi obtida com um EXPERT 1.1 e há casos em que outros tipos de MSX produzem somas diferentes.

Esperamos que este livro possa abrir novos horizontes aos programadores MSX, tanto aos iniciantes quanto aos que não têm tempo suficiente para descobrir sozinhos os incontáveis "macêtes" dessas máquinas maravilhosas.



## DICAS SOBRE O TECLADO

As dicas deste capítulo abordam preponderantemente assuntos relativos a manipulação do teclado nos micros MSX.

Uma vez que o principal meio de entrada de dados para a UCP do micro é o teclado, a utilidade destas dicas é evidente.

1.1 - Carregando o Buffer do Teclado .....	8
1.2 - Limpando o Buffer do Teclado .....	9
1.3 - Programando as teclas de funções .....	10
1.4 - Restabelecendo as teclas de funções ..	11
1.5 - Checando as teclas especiais .....	12
1.6 - Travando a tecla CAPS LOCK .....	13
1.7 - Usando o click do teclado .....	14
1.8 - Reprogramando todo o teclado .....	15
1.9 - Usando a barra de espaços .....	16
1.A - Usando as teclas de setas .....	19
1.B - Uso da instrução ON KEY GOSUB .....	20

## 1.1 - CARREGANDO O BUFFER DO TECLADO

O MSX reserva uma área (buffer) de 40 bytes na memória RAM para armazenar temporariamente os dados digitados através do teclado. Se esses dados forem comandos, serão executados na sequência em que estiverem no buffer. Pode-se carregar comandos no buffer do teclado através de programas em BASIC ou em ASSEMBLY a fim de gerar efeitos especiais. Por exemplo, para fazer com que um programa em BASIC seja carregado de fita cassete e seja automaticamente executado, basta usar o programa apresentado a seguir.

100	SCREEN 1:X\$=CHR\$(34):Y\$=CHR\$(13)	200
110	A\$="CLOAD"+X\$+"CAS:"+X\$+"=RUN"+Y\$	159E
120	EN=&HFBF0	1917
130	FOR F=1 TO LEN(A\$)	208E
140	CH=ASC(MID\$(A\$,F,1))	250F
150	POKE EN,CH	2018
160	EN=EN+1	3117
170	NEXT F	3407
180	X=45536!+&HFBF0+LEN(A\$)	4088
190	Y=X-256*INT(X/256)	440F
200	POKE &HF3F8,Y	4F80
210	POKE &HF3F9,INT(X/256)	5890
220	POKE &HF3FA,&HF0	5E43
230	POKE &HF3FB,&HF0	6407

TOTAL = 6407

Os dados a serem inseridos no buffer estão na variável A\$. A variável EN armazena o endereço do buffer a ser preenchido. A variável CH armazena o caractere de A\$ a ser inserido no endereço EN do buffer. O endereço &HFBF0 é o início do buffer do teclado (KEYBUFF).

Os endereços &HF3F8 e &HF3F9 armazenam o próximo endereço a ser preenchido no buffer do teclado (PUTPNT) e são preenchidos de modo a apontarem para o endereço subsequente ao do fim da mensagem inserida.

Os endereços &HF3FA e &HF3FB armazenam o último endereço do buffer lido pelo micro (GETPNT) e são posicionados de modo a apontarem para o endereço do primeiro caractere inserido (&HFBF0).

## BIBLIOGRAFIA RECOMENDADA:

Programação Avançada em MSX - páginas 40, 41, 42 e 143.

## 1.2 - LIMPANDO O BUFFER DO TECLADO

Observe o programa a seguir:

```
10 SCREEN 0,,0:LOCATE,,0:KEY OFF
20 PRINT " Entre uma escolha:":PRINT
30 PRINT " [ 1 ] Opção 1 ...."
40 PRINT " [ 2 ] Opção 2 ...."
50 PRINT " [ 3 ] Opção 3 ...."
60 PRINT:PRINT:PRINT
70 B$=INKEY$:IF LEN(B$)<1 THEN 70
80 FOR F=1 TO 500 : NEXT F : BEEP
90 PRINT "OPÇÃO ";B$;" ESCOLHIDA !"
100 A$=INPUT$(1)
110 PRINT A$
```

1002  
1033  
1441F  
1E5A  
2176  
2884  
3802  
4407  
5800  
5F15  
6378

TOTAL = 6378

Note que ao fazer a primeira opção, se a tecla da escolha for pressionada por muito tempo ou se mais de uma tecla for pressionada, a segunda opção também será feita. Isso ocorre porque o buffer do teclado fica carregado com os caracteres digitados até que eles sejam usados.

No BIOS do MSX existe uma rotina que pode ser útil nessas situações: a KILLBUFF. Sempre que é executada, ela limpa o buffer do teclado. Para chamá-la, basta usar as instruções:

```
DEFUSR0 = &H0156 : POKE 0,USR0(0)
```

Experimente inserir a linha a seguir no programa anterior e depois execute-o novamente.

```
95 DEFUSR0 = &H0156 : POKE 0,USR0(0)
```

Você notará que a segunda opção não mais será atrapalhada por digitações acidentais.

### BIBLIOGRAFIA RECOMENDADA:

Coleção de Programas para MSX vol. 2 - página 110.  
Aprofundando-se no MSX - página 159.



### 1.3 - PROGRAMANDO AS TECLAS DE FUNÇÕES

As teclas de funções podem ser facilmente redefinidas para atender às necessidades específicas de um programa.

A forma mais direta de reprogramá-las é usando a instrução KEY. Entretanto pode-se também redefinir as teclas de funções diretamente na memória RAM.

Experimente executar o seguinte programa e depois pressione a tecla F1.

```
10 FOR F=&HF87F TO &HF87F+3B
20   POKE F,ASC("A")
30 NEXT F
```

715  
830  
848

Com isso a tecla F1 passou a ter uma sequência de 39 caracteres "A". Lembre-se que com a instrução KEY podemos inserir no máximo 15 caracteres em cada tecla de função. Com este recurso, podemos atribuir a uma única tecla até 39 caracteres, entretanto deve-se tomar alguns cuidados, pois o conteúdo das demais teclas podem ser alterados. Experimente digitar a tecla F2 após ter executado o programa acima. Você verá que seu conteúdo foi alterado pelo programa.

O que acontece é que existem 160 bytes da RAM, reservados a partir do endereço &HF87F (FNKSTR), para armazenar os textos das teclas de funções. Cada tecla tem seu texto começando sempre num mesmo endereço e o número máximo de caracteres atribuíveis a uma única tecla de função é 39. O 40º caractere da área de texto de uma tecla é sempre um 0.

Você pode também atribuir às teclas de funções sequências de caracteres de controle. Por exemplo, digite a instrução a seguir e depois pressione a tecla F1.

```
KEY 1,CHR$(7)+CHR$(28)+CHR$(8)+CHR$(7)+CHR$(127)+CHR$(9)+CHR$(11)
```

### BIBLIOGRAFIA RECOMENDADA:

Linguagem Basic MSX - página 81.  
Curso de Basic v.1 - páginas 21 e 22.  
Aprofundando-se no MSX - página 50.  
Programação Avançada em MSX - página 145

## 1.4 - RESTABELECENDO AS TECLAS DE FUNÇÕES

As teclas de funções podem ser facilmente programadas. Entretanto, após usar um programa que as redefine, pode ser necessário reinicializá-las com as funções originais. Para isso pode-se simplesmente programar tecla por tecla novamente ou chamar uma rotina do BIOS (INITFNK em &H003E) que se encarrega de fazer isso automaticamente. Para executar essa rotina do BIOS basta digitar a seguinte instrução:

```
DEFUSR0 = &H3E : POKE 0,USR0(0)
```

Observe o programa exemplo a seguir. Ele redefine as teclas de funções e logo a seguir restabelece seus conteúdos originais.

10 SCREEN 0 : KEY ON	000
20 FOR F=1 TO 10	002
30 KEY F,"NOVA !"	007
40 NEXT F	0FE
50 SCREEN 0	065
60 PRINT,," TECLAS REDEFINIDAS !"	1390
70 PRINT,," PRESSIONE RETURN !"	1307
80 AS=INPUT\$(1)	1000
90 DEFUSR0 = &H3E : X = USR0(0)	23F0
100 SCREEN 0	20FF
110 PRINT,," TECLAS RESTABELECIDAS !"	4646

TOTAL = 4646

## BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX - página 156.

## 1.5 - CHEGANDO AS TECLAS ESPECIAIS

Nos micros MSX existem várias maneiras de verificarmos se alguma tecla especial está ou não pressionada. Uma delas é consultando as variáveis do sistema OLDKEY (&HFBDA) ou NEWKEY (&HFBEE5). O programa listado a seguir ilustra esse procedimento, checando especificamente se a tecla LGRA (ou GRAPH) está pressionada.

Ao ser executado o programa lê o conteúdo da região da memória RAM que vai do endereço &HFBDA até o endereço &HFBEEF, mostrando na tela seus conteúdos em binário. Experimente pressionar algumas teclas (menos ^STOP!). Você verá que a cada tecla pressionada corresponde um bit de algum dos bytes dessa região. Para checar qualquer tecla, portanto, basta verificar se o bit está ou não em "0". Experimente pressionar a tecla LGRA (ou GRAPH). A configuração que corresponde a ela é o valor 251 (ou 11111011, em binário) no byte &HFBEE0. Observe como a linha 250 faz o teste para ver se ela está ou não pressionada e tente alterar o programa para que ele teste se a tecla RGRA (ou CODE) está pressionada.

```
100 REM                                     F3
110 REM LE OLDKEY E NEWKEY               F3E
120 REM                                   84F
130 SCREEN 0                             82
140 LOCATE 0,5,0                         82
150 FOR F=&HFBDA TO &HFBEE4             48F
160   PRINT "&H";HEX$(F);"<=> ";      100
170   PRINT RIGHT$("00000000"+BIN$(PEEK(F)),8);
180   PRINT "   &H";HEX$(F+1);"<=> ";  82E
190   PRINT RIGHT$("00000000"+BIN$(PEEK(F+1)),8) 4EE
200 NEXT F                               400
210 PRINT:PRINT:PRINT                   84E
220 REM                                  85E
230 REM TESTA A TECLA LGRA             860
240 REM                                  86E
250 IF PEEK(&HFBEE0)=(PEEK(&HFBEE0)AND251)AND251 THEN PRINT "LGRA PRESSIONADA"ELSEPRINT"
260 GOTO 140                             800
```

### BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - páginas 83 a 86.

## 1.6 - TRAVANDO A TECLA CAPS LOCK

Existe uma variável do sistema (CAPST, em &HFCAB) que indica o estado da tecla CAPS LOCK. Se CAPS LOCK está ativa, essa variável contém 255 ou algum outro valor maior que zero, e se inativa, contém 0. Cada vez que a tecla CAPS LOCK é pressionada os bits da variável CAPST são complementados de modo que seu valor é sempre 0 ou 255.

Para simular o pressionamento de CAPS LOCK basta "pokear" 255 ou 0 em CAPST.

Para travar a CAPS LOCK, deixando-a ativa, basta pokear qualquer valor maior que 0 e menor que 255 em CAPST, pois assim, mesmo que CAPS LOCK seja pressionada, a complementação dos bits de CAPST produzirá um valor também maior que 0 e menor que 255.

Experimente digitar e usar o programa a seguir. Após executá-lo, digite algumas letras usando as teclas CAPS LOCK e SHIFT. Você notará que elas ficaram praticamente inoperantes para as letras.

```
10 SCREEN 0 : WIDTH 38
20 POKE &HFCAB,1
30 PRINT,"      DIGITE ALGUMAS LETRAS"
40 PRINT,"      (com ou sem SHIFT):"
50 PRINT : PRINT
60 A$=INPUT$(1)
70 PRINT A$;
80 GOTO 60
```

255  
255  
194  
194  
194  
194  
194  
2023  
2023  
2023

TOTAL : 2023

Para fazer com que a tecla CAPS LOCK volte a funcionar normalmente, comande:

POKE &HFCAB,0

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - página 50.  
Programação Avançada em MSX - página 148.  
Coleção de Programas para MSX vol 2 - páginas 51 e 55.

## 1.7 - USANDO O CLICK DO TECLADO

O click do teclado pode ser usado para muitas e diferentes aplicações, deixando livre para outros usos o PSG.

Para gerar o click do teclado por software, é necessário acessar o hardware da máquina com comandos OUT. Veja o programa a seguir.

10	SCREEN 1	10E
20	KEY OFF	2F5
30	COLOR 1,4	451
40	LOCATE 32*RND(1),24	941
50	PRINT "*"	A44
60	OUT &HAA,&HFF	1056
70	OUT &HAA,&H7F	1091
80	GOTO 40	1363

TOTAL = 1363

Cada vez que as linhas 60 e 70 são executadas um "click" é gerado. Você pode usar isto para sonorizar seus programas mesmo sem usar o PSG.

## BIBLIOGRAFIA RECOMENDADA:

Coleção de Programas para MSX vol 2 - página 15.  
Aprofundando-se no MSX - página 87.

## 1.8 - REPROGRAMANDO TODO O TECLADO

Agora seu MSX também pode ter um "teclado inteligente" como nos velhos tempos do Apple e do Sinclair. O programa a seguir instala em seu teclado as principais palavras reservadas do Basic sem perder sua função original.

A redefinição atinge as letras maiúsculas e minúsculas, totalizando 52 palavras reservadas. No que é possível, as teclas seguem o padrão Sinclair (A=NEW, K=LIST, etc). Palavras do Sinclair que não existem no MSX (PAUSE, SCROLL, etc) são trocadas por outras do último.

Para usar o programa, digite-o e comande RUN. Após a execução deverão aparecer os códigos das teclas e as palavras correspondentes. Experimente então teclar SELECT e em seguida a tecla A. Você deverá obter "NEW". Tecle agora SELECT+"a" para obter "FRE(". Tente então outras combinações. O que acontece quando a tecla após o SELECT não é uma letra?

```

1000 REM -----
1010 REM  TOKEN V. 1.0 BY THE PILOT
1020 REM      JANEIRO DE 1988
1030 REM -----
1040 REM
1050 DATA 21,9B,D1,CD,CE,D1,21,16
1060 DATA D0,22,A5,FD,3E,C3,32,A4
1070 DATA FD,AF,32,D7,D1,C9,4F,3A
1080 DATA D7,D1,A7,79,20,05,FE,1B
1090 DATA 2B,49,C9,FE,41,38,3D,FE
1100 DATA 5B,30,08,D6,40,47,21,72
1110 DATA D0,18,0E,FE,61,38,2D,FE
1120 DATA 7B,30,29,D6,60,47,21,0B
1130 DATA D1,7E,A7,23,20,FB,10,F9
1140 DATA 3E,C9,32,A4,FD,7E,A7,28
1150 DATA 06,CD,A2,00,23,18,F6,3E
1160 DATA C3,32,A4,FD,AF,32,D7,D1
1170 DATA C1,C3,DA,08,4F,AF,32,D7
1180 DATA D1,79,C9,3E,FF,32,D7,D1
1190 DATA AF,C9,00,4E,45,57,00,42
1200 DATA 45,45,50,00,43,4F,4E,54
1210 DATA 00,44,49,4D,20,00,52,45
1220 DATA 4D,20,00,46,4F,52,20,00
1230 DATA 47,4F,54,4F,20,00,47,4F
1240 DATA 53,55,42,20,00,49,4E,50
1250 DATA 55,54,20,00,4C,4F,41,44
1260 DATA 20,00,4C,49,53,54,20,00

```

```

246
FTE
1509
2051
222E
30E9
4300
5522
6202
8056
88E0
8E70
9600
9EEE
A74F
B3F9
CB32
DEFC
E3CF
F96
94B
C60
17FE
2173
3003
41A6
5876

```

1270 DATA 4C,4C,49,53,54,20,00,4D	0000
1280 DATA 4F,54,4F,52,20,00,4E,45	0001
1290 DATA 58,54,20,00,50,4F,48,45	0002
1300 DATA 20,00,50,52,49,4E,54,00	0003
1310 DATA 50,53,45,54,20,28,00,52	0004
1320 DATA 55,4E,00,53,41,56,45,20	0005
1330 DATA 00,54,52,4F,4E,00,49,46	0006
1340 DATA 20,00,43,4C,53,00,50,52	0007
1350 DATA 45,53,45,54,20,28,00,43	0008
1360 DATA 4C,45,41,52,00,52,45,54	0009
1370 DATA 55,52,4E,00,45,4E,44,00	0010
1380 DATA 00,46,52,45,28,00,49,4E	0011
1390 DATA 48,45,59,24,00,44,53,48	0012
1400 DATA 46,28,00,41,54,4E,28,00	0013
1410 DATA 54,41,4E,28,00,53,47,4E	0014
1420 DATA 28,00,41,42,53,28,00,53	0015
1430 DATA 51,52,28,00,41,53,43,28	0016
1440 DATA 00,56,41,4C,28,00,4C,45	0017
1450 DATA 4E,28,00,55,53,52,00,33	0018
1460 DATA 2E,31,34,31,35,39,32,37	0019
1470 DATA 21,00,4E,4F,54,00,50,45	0020
1480 DATA 45,48,28,00,54,41,42,28	0021
1490 DATA 00,53,49,4E,28,00,49,4E	0022
1500 DATA 54,28,00,53,54,52,49,4E	0023
1510 DATA 47,24,28,00,52,4E,44,28	0024
1520 DATA 00,43,48,52,24,28,00,56	0025
1530 DATA 41,52,50,54,52,28,00,43	0026
1540 DATA 4F,53,28,00,45,58,50,28	0027
1550 DATA 00,53,54,52,24,28,00,4C	0028
1560 DATA 4E,28,00,0C,50,72,6F,67	0029
1570 DATA 72,61,6D,61,20,65,73,63	0030
1580 DATA 72,69,74,6F,20,70,6F,72	0031
1590 DATA 3A,0D,0A,54,48,45,20,50	0032
1600 DATA 49,4C,4F,54,20,65,6D,20	0033
1610 DATA 4A,61,6E,65,69,72,6F,2F	0034
1620 DATA 31,39,38,38,2E,00,7E,A7	0035
1630 DATA C8,CD,A2,00,23,18,F7,00	0036
1640 DATA FIM	0037
1650 CLS:PRINT "CARREGANDO TOKEN"	0038
1660 FOR I=&HD000 TO &HD1D8:READ A\$:POKE	0039
I,VAL("&H"+A\$):NEXT I	0040
1670 DEFUSR=&HD000:A=USR(0):PRINT:PRINT	0041
1680 FOR I=65 TO 90:PRINT "<SELECT>+"CHR	0042
\$(I);" = ";CHR\$(24);CHR\$(I):FOR T=0 TO 1	0043
00:NEXT T:NEXT I	0044
1690 FOR I=97 TO 122:PRINT "<SELECT>+"CHR	0045
R\$(I);" = ";CHR\$(24);CHR\$(I):FOR T=0 TO	0046
100:NEXT T:NEXT I	0047
1700 END	0048

É recomendável salvar o código binário para facilitar o uso do programa. Para isto digite:

```
BSAVE "TOKEN.BIN",&HD000,&HD1DB
```

EU

```
BSAVE "CAS:TOKEN",&HD000,&HD1D8
```

**Para executar o programa, use o comando:**

LOAD "TOKEN.BIN",R

1

BLOAD "CAS:".R

UNITED STATES OF AMERICA: RECEIVED SEP 24 1964

Aprofundando-se no MSX - páginas 66 a 71 e capítulo 3.  
Programação Avançado em MSX - capítulo 1 e apêndice 3.



## 1.9 - USANDO A BARRA DE ESPAÇOS

A barra de espaços no MSX pode ser usada de muitas e diferentes maneiras. No programa abaixo apresentamos um exemplo em que a barra é usada para controlar o fluxo do processamento usando a função STRIG.

```
10 SCREEN 0 : PRINT : PRINT SPC(6);  
20 PRINT "DIGITE A BARRA DE ESPAÇO !"  
30 BEEP  
40 IF NOT STRIG(0) THEN 30  
50 PLAY "ABCDEFGG"  
  
TOTAL = 1914
```

Uma outra forma de se controlar o fluxo do processamento é através das interrupções. Essa maneira é ilustrada pelo programa a seguir.

```
10 SCREEN 0 : PRINT : PRINT SPC(6);  
20 PRINT "DIGITE A BARRA DE ESPAÇO !"  
30 STRIG(0) ON : ON STRIG GOSUB 60  
40 PLAY "V15ABCDEFGG07L32"  
50 GOTO 50  
60 PLAY "CH" : RETURN  
  
TOTAL = 2318
```

Observe que os dois programas apresentados são fundamentalmente diferentes. O primeiro apenas interrompe o processamento normal do programa até que a barra de espaços seja pressionada, enquanto o segundo desvia o processamento para uma sub-rotina, esteja ele em que linha estiver, sempre que a barra de espaços for pressionada.

## 1.A - USANDO AS TECLAS DE SETAS

As teclas de setas podem ser testadas através de uma função do BASIC MSX. O programa apresentado a seguir ilustra um uso típico dessa função, associada ao controle de um ponto plotante na SCREEN 2.

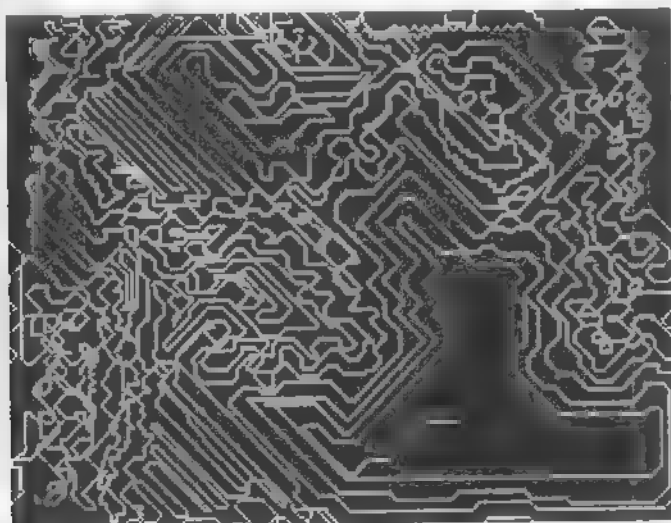
Note que usamos "parênteses lógicos" para tornar a tela ilimitada. Se o ponto tentar sair da tela por qualquer um de seus 4 lados, automaticamente será transportado para a primeira posição do lado oposto.

```
100 SCREEN 2,,0 : DEFINT A-Z
110 X=128 : Y=96
120 PSET (X,Y),15
130 A=STICK(0) : IF A=0 THEN 130
140 Y=Y+(A<3 OR A=8)-(A>3 AND A<7)
150 X=X-(A>1 AND A<5)+(A>5)
160 X = -255*(X=-1) - X*(X<>-1)
170 X = -X*(X<>256)
180 Y = -191*(Y=-1) - Y*(Y<>-1)
190 Y = -Y*(Y<>192)
200 GOTO 120
```

48C  
99C  
C74  
14F6  
2178  
2ADA  
3A22  
43C8  
51D8  
59B4  
6E2E

TOTAL = 5EDE

Para entender o programa, lembre-se que quando o conteúdo dentro dos parênteses for verdadeiro, ele pode ser substituído pelo valor numérico "-1", e quando for falso, pelo valor "0".



## 1.8 - USO DA INSTRUÇÃO ON KEY GOSUB

As teclas de funções podem ser usadas para chamar sub-rotinas automaticamente durante as interrupções. Observe o programa a seguir. Digite-o e rode-o.

10 SCREEN 0,,0 : PLAY "L64"	851
20 FOR F=1 TO 10 : KEY(F) ON : NEXT F	887
30 ON KEY GOSUB 100,200,300,400,500,600,700,800,900,1000	1016
40 GOTO 40	1128
100 PRINT " F1 PRESSIONADA !!! "	21AC
110 PLAY "C"	282E
120 RETURN	2C89
200 PRINT " F2 PRESSIONADA !!! "	3EE0
210 PLAY "CH"	4378
220 RETURN	4488
300 PRINT " F3 PRESSIONADA !!! "	5875
310 PLAY "D"	5D9E
320 RETURN	5F5F
400 PRINT " F4 PRESSIONADA !!! "	72FE
410 PLAY "DH"	7891
420 RETURN	7A54
500 PRINT " F5 PRESSIONADA !!! "	839A
510 PLAY "E"	8670
520 RETURN	8740
600 PRINT " F6 PRESSIONADA !!! "	8E5F
610 PLAY "F"	9101
620 RETURN	923B
700 PRINT " F7 PRESSIONADA !!! "	9AB4
710 PLAY "FH"	9D7E
720 RETURN	9F86
800 PRINT " F8 PRESSIONADA !!! "	ABE9
810 PLAY "G"	AD2C
820 RETURN	AE7E
900 PRINT " F9 PRESSIONADA !!! "	C088
910 PLAY "GH"	C4E8
920 RETURN	C5F7
1000 PRINT " F10 PRESSIONADA !!! "	D798
1010 PLAY "A"	DC10
1020 RETURN	1060

TOTAL = 0060

Observe que, independentemente da linha do programa que estiver sendo executada, sempre que alguma tecla de função for pressionada uma sub-rotina será chamada.



## DICAS PARA USAR O VIDEO

Este capítulo aborda os recursos do vídeo do MSX. Existe um circuito dedicado ao controle do vídeo (VDP) que tem à sua disposição 16 Kbytes de memória RAM (VRAM) para armazenar os dados da tela. O controle do vídeo pode ser feito através do BASIC, com os vários comandos dedicados a isso, ou diretamente em Linguagem de Máquina.

2.1 - Cor de frente igual a cor de fundo ....	22
2.2 - Pseudo-borda na SCREEN 0 ....	23
2.3 - Textos na SCREEN 2 .....	24
2.4 - Caracteres menores que 32 .....	26
2.5 - Caracteres de controle do vídeo .....	27
2.6 - Usando o STEP em comandos gráficos ....	29
2.7 - Posicionando o cursor na SCREEN 2 .. .	31
2.8 - Movimentos na tela .....	32
2.9 - SCREEN 4 .....	35
2.A - Setores com o CIRCLE .....	39
2.B - Redefinindo caracteres .....	40
2.C - SCROLL UP para SCREEN 0 .....	52
2.D - SCROLL DOWN para SCREEN 0 .....	53
2.E - SCROLL LEFT para SCREEN 0 .....	54
2.F - SCROLL RIGHT para SCREEN 0 .....	55
2.G - SCROLL UP para SCREEN 1 .....	56
2.H - SCROLL DOWN para SCREEN 1 .....	57
2.I - SCROLL LEFT para SCREEN 1 .....	58
2.J - SCROLL RIGHT para SCREEN 1 .....	59
2.K - Centralizando caracteres .....	60
2.L - Animação com SPRITES .....	61
2.M - Letras ampliadas .....	63
2.N - Entendendo o DRAW .....	66
2.O - "WARP 8" na SCREEN 2 .....	67
2.P - "SPRITEANDO" a tabela de caracteres ...	68
2.Q - Arlequim bêbado .....	69
2.R - Usando 40 ou 64 colunas na SCREEN 2 ...	71
2.S - Carimbador de SPRITES 8x8 na SCREEN 2 .	73
2.T - Carimbador de SPRITES 16x16 na SCREEN 2	75
2.U - Armazenando telas na RAM .....	76
2.V - Usando a VRAM para dados .....	77
2.W - Pesquisador de desenhos .....	79
2.X - Imagens instantâneas .....	83
2.Y - Impressão em tamanho duplo na SCREEN 2	84

## 2.1 - COR DE FRENTE IGUAL A COR DE FUNDO

Muitos programas ao terminarem a execução ou serem interrompidos por CONTROL + STOP deixam a tela com a cor de frente igual a cor de fundo e a primeira impressão que se tem é que o micro quebrou. Digite e rode o programinha listado a seguir:

```
10 SCREEN 2
20 CIRCLE (128,84),50
30 COLOR 1
40 GOTO 40
```

```
100
200
300
400
500
```

Após o desenho do círculo terminar, digite CONTROL + STOP. A tela deverá ficar totalmente escura.

Para verificar o que está acontecendo basta digitar:

```
SHIFT + HOME/CLS    e
SHIFT + F1
```

Com isso, a tela será limpa e as cores normais do vídeo serão restabelecidas, desde que as teclas de funções não tenham sido redefinidas pelo programa que foi interrompido.

Se o procedimento descrito acima não funcionar, tente digitar o comando abaixo, mesmo sem vê-lo na tela:

```
COLOR 15,1 : SCREEN 0 (e RETURN)
```

## 2.2 - PSEUDO-BORDA NA SCREEN 0

O comando COLOR, quando usado com a SCREEN 0, não permite a especificação da cor da borda. Com um pequeno programa em BASIC podemos resolver parcialmente o problema, gerando uma PSEUDO-BORDA para a SCREEN 0. Digite e execute o programa a seguir e depois verifique as novas características da tela.

10 COLOR 1,15 : SCREEN 0	208
20 FOR F=2048 TO 4095	784
30   X = NOT(VPEEK(F)) AND 255	DC8
40   VPOKE F, X	105F
50 NEXT F	110F
60 INPUT"Qual a cor da borda (0-15)";B	18FC
70 IF B<0 OR B>15 THEN 60	2420
80 COLOR ,B	261F
90 GOTO 60	2A84

TOTAL = 2A84

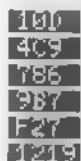
### BIBLIOGRAFIA RECOMENDADA

Aprofundando-se no MSX - capítulo 4.

## 2.3 - TEXTOS NA SCREEN 2

O MSX permite a impressão de letras e gráficos na SCREEN 2. Para isso é necessário abrir um arquivo na tela (GRP:) e usar a instrução PRINT #. Observe o programa a seguir:

```
10 SCREEN 2
20 OPEN "GRP:" AS #1
30 CIRCLE (128,86),50
40 PRESET (110,84)
50 PRINT #1, "EDITORA ALEPH"
60 GOTO 60
```



A linha 10 seleciona o uso da tela gráfica de alta resolução (GRP:).

A linha 20 abre um arquivo nessa tela. Não é necessário especificar o tipo de arquivo ( ...FOR OUTPUT...), uma vez que ele só pode ser de saída.

A linha 30 desenha um círculo de centro na posição (128,86) e raio de 50 pixels.

A linha 40 "marca" um ponto na posição (110,84), a partir do qual a mensagem será impressa.

A linha 50 imprime a mensagem na tela a partir do ponto marcado pela linha 40. Cada caractere é definido dentro de uma matriz de 8x8 pontos. A posição marcada pela linha 40 posiciona o vértice superior esquerdo do primeiro caractere da mensagem.

Observe que podemos "criar" novos tipos de letras na SCREEN 2, usando uma dupla impressão dos caracteres normais. Experimente inserir no programa anterior as seguintes linhas:

```
55 PRESET (109,84)
56 PRINT #1, "EDITORA ALEPH"
```

Isso deve ter produzido uma mensagem em "bold".

Um outro recurso é o uso de espaçamento menor entre as letras a serem impressas. Experimente executar o programa a seguir. Ele imprime na SCREEN 2 uma mensagem com espaçamento reduzido.

```
10 SCREEN 2
20 OPEN "GRP:" AS #1
30 AS="EDITORA ALEPH"
```



```

40 FOR F=1 TO LEN(AS)
50 PRESET (110+(F-1)*6,84)
60 PRINT #1,MID$(AS,F,1)
70 NEXT F
80 GOTO 80

```

```

100E
178E
118E
215C
2489

```

100HL = 2489

Para entender melhor o funcionameneto do programa, experimente substituir a linha 50 por:

```

50 PRESET (110+(F-1)*12,84)

```



#### BIBLIOGRAFIA RECOMENDADA:

Curso de BASIC v.1 - páginas 65 e 66.  
 Coleção de Programas para MSX - páginas 32, 33, 59 e 60.



## 2.4 - CARACTERES MENORES QUE 32

Os micros MSX dispõem de 256 caracteres, todos apresentáveis no vídeo. Entretanto, os primeiros 32 caracteres correspondem a códigos de controle de periféricos (0 a 31) e para serem apresentados através de seus códigos necessitam de uma sintaxe peculiar da instrução PRINT CHR\$. O programa a seguir apresenta os 32 caracteres de controle através de seus códigos.

```
10 SCREEN 1
20 WIDTH 16
30 FOR F=0 TO 31
40 PRINT CHR$(1)+CHR$(64+F);
50 NEXT F
```

```
10E
21E
59F
M1C
E99
```

Note que a apresentação é feita pela linha 40. Para se mostrar um caractere de controle através de seu código é necessário usar a sintaxe:

```
PRINT CHR$(1)+CHR$(64+ n2 )
```

Onde "n2" é o código (de 0 a 31) do caractere a ser apresentado.



## BIBLIOGRAFIA RECOMENDADA:

Coleção de Programas para MSX v.1 - páginas 26 e 27.

## 2.5 - CARACTERES DE CONTROLE DE VÍDEO

O padrão de vídeo dos MSX é um dos mais utilizados em microcomputadores. Alguns caracteres podem ser usados para controlar o vídeo em funções como o posicionamento do cursor, limpeza da tela, formato do cursor, etc.

A seguir relacionamos os caracteres de controle do vídeo do MSX. Eles devem ser usados com o comando PRINT. Por exemplo, para deixar o cursor posicionado na linha 10 e coluna 20 basta comandar:

```
PRINT CHR$(27)CHR$(32+10)CHR$(32+20);
```

**Apaga a tela**

```
CHR$(27)"J"      ou  
CHR$(27)"E"
```

**Apaga do cursor ao fim da linha**

```
CHR$(27)"K"
```

**Apaga do cursor ao fim da tela**

```
CHR$(27)"J"
```

**Apaga a linha inteira**

```
CHR$(27)"I"
```

**Insere uma linha em branco**

```
CHR$(27)"L"
```

**Elimina uma linha**

```
CHR$(27)"M"
```

**Posiciona cursor**

```
CHR$(27)"Y";  
CHR$(32+nº da linha);  
CHR$(32+nº da coluna)
```

**Cursor linha acima**

```
CHR$(27)"A"
```

**Cursor linha abaixo**

```
CHR$(27)"B"
```

**Cursor coluna a direita**

```
CHR$(27)"C"
```

**Cursor coluna a esquerda**

```
CHR$(27)"D"
```

**Cursor em HOME**  
CHR\$(27)"H"

**Cursor inteiro**  
CHR\$(27)"x4"

**Cursor pela metade**  
CHR\$(27)"y4"

**Cursor apagado**  
CHR\$(27)"x5"

**Cursor aceso**  
CHR\$(27)"y5"

O programa a seguir ilustra o uso de alguns dos recursos descritos acima. Digite-o, rode-o e estude-o.

10 SCREEN 0 : WIDTH 38 : KEY OFF	0000
11 PRINT "Digite qualquer coisa ";	0056
12 PRINT "e use as teclas de setas";	0108
13 PRINT " e as teclas HOME e CLS !"	0160
14 PRINT : PRINT : PRINT	0212
20 A\$=INKEY\$ : IF A\$="" THEN 20	0264
30 IF A\$(<)CHR\$(12) THEN 50	0316
40 PRINT CHR\$(27)"j"	0368
50 IF A\$(<)CHR\$(11) THEN 70	0420
60 PRINT CHR\$(27)"k"	0472
70 PRINT A\$;	0524
90 GOTO 20	0576

#### BIBLIOGRAFIA RECOMENDADA.

Sistema de Disco para MSX - página 186.

## 2.6 - USANDO O STEP EM COMANDOS GRÁFICOS

A instrução auxiliar STEP pode ser muito útil quando usada com comandos gráficos.

Veja o programa a seguir:

```
10 SCREEN 2
20 PSET (20,20)
30 PSET STEP (20,20)
40 GOTO 40
```

100  
200  
300  
400

500 600 700 800 900

Ao ser executado ele irá plotar dois pontos na diagonal que sai do canto superior esquerdo da tela.

Isso acontece porque a instrução STEP permite a definição de um novo sistema de coordenadas na tela.

O sistema de coordenadas normal é um sistema ABSOLUTO, isto é, suas coordenadas são sempre correspondentes a uma mesma origem, fixa no canto superior esquerdo da tela.

O sistema usado pelo STEP é um sistema MÓVEL, em que as coordenadas correspondem a uma origem móvel, definida pelo último ponto "marcado" na tela. Note que PONTO "MARCADO" não é necessariamente PONTO "PLOTADO". Por exemplo, ao se usar um comando CIRCLE o ponto "marcado" é o seu centro, enquanto que os pontos plotados são os do círculo.

Vamos tentar entender isso melhor.

Observe novamente o programa anterior.

A linha 10 apenas seleciona a SCREEN 2 do micro. Essa tela tem uma resolução de 256 colunas x 192 linhas, num total de 49152 pontos.

A linha 20 marca o ponto de coordenadas  $x=20$  e  $y=20$  na tela. Note que essas coordenadas são referentes ao sistema ABSOLUTO, pois NÃO existe a instrução STEP precedendo as coordenadas.

A linha 30 também marca um ponto na tela (o de coordenadas  $x'=20$  e  $y'=20$ ), porém as coordenadas desse ponto são precedidas pela instrução STEP e, portanto, referem-se ao sistema de coordenadas MÓVEL. Como o último ponto marcado na tela foi o de coordenadas ABSOLUTAS  $x=20$  e  $y=20$ , esse ponto foi tomado como ORIGEM do sistema MÓVEL. Portanto, as coordenadas  $x'=20$  e  $y'=20$  do sistema MÓVEL correspondem às coordenadas  $x=40$  e  $y=40$  do sistema ABSOLUTO.

Agora, esquite um pouco a cabeça tentando entender o funcionamento dos três programinhas listados na próxima página.

# FIGURE 1

```
10 SCREEN 2
20 PSET (0,0)
30 CIRCLE STEP(6,6),4
40 GOTO 30
```

100

288

574

705

# FIGURE 2

```
10 SCREEN 2
20 PSET (0,0)
30 LINE STEP(3,3)-STEP(3,3)
40 GOTO 30
```

100

288

740

874

# FIGURE 3

```
10 SCREEN 2
20 FOR F=0 TO 6.28 STEP .2
30   X=80*SIN(F) : Y=80*COS(F)
40   LINE STEP(X,Y)-(128,86)
50 NEXT F
60 GOTO 60
```

100

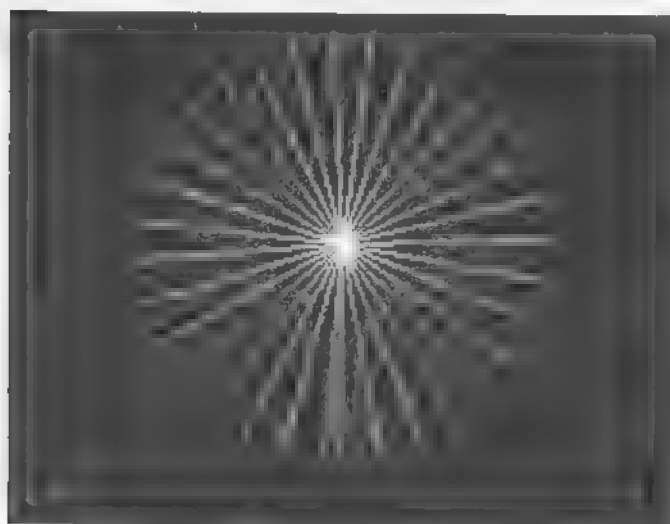
702

1100

1868

1485

104E



## 2.7 - POSICIONANDO O CURSOR NA SCREEN 2

A forma mais genérica de se posicionar o cursor na SCREEN 2 é usando o seguinte comando:

```
PSET (coluna,linha),POINT (coluna,linha)
```

Por exemplo, para posicionar o cursor na coluna 89 e linha 190 devemos fazer:

```
PSET (89,190),POINT (89,190)
```

Se quisermos posicionar o cursor para impressão de textos na SCREEN 2 podemos considerá-la dividida em 32 colunas e 24 linhas, como a SCREEN 1. Desse modo, o comando PSET acima pode ser substituído por:

```
PSET (8*col,8*lin),POINT (8*col,8*lin)
```

Para imprimir na coluna 10 e linha 5 devemos usar então:

```
PSET (8*10,8*5),POINT (8*10,8*5)
```

Agora, digite e rode o programa a seguir:

```
10 SCREEN 2
20 OPEN "GRP:" AS #1
30 C = 10 : L = 10
40 PSET(8*C,8*L),POINT(8*C,8*L)
50 PRINT #1, "ALEPH"
60 GOTO 50
```

```
100
4C9
8CE
1108
1150C
181E
```

```
101ML = 181E
```

Na linha 30 define-se a coluna (de 0 a 31) e a linha (de 0 a 23) onde desejamos imprimir. A linha 40 posiciona o cursor e a linha 50 imprime na tela.

Veja a dica 2.3 para entender melhor o programa.

## 2.8 - MOVIMENTOS NA TELA

Muitas vezes pode-se desejar que a tela do MSX apresente algum movimento global. A seguir, apresentamos seis programas que produzem movimentos globais na SCREEN 1.

### PROGRAMA 1

```
10 ' TESTE DE MOVIMENTO
20 SCREEN 1 : WIDTH 32 : COLOR 1,7,4
30 FOR F=0 TO 22
40 PRINT STRING$(32,"0"); : NEXT F
50 FOR F=8*ASC("0") TO 8*ASC("0")+7
60 IF B=0 THEN VPOKE F,VPEEK(F)/8
70 IF B=1 THEN VPOKE F,8*VPEEK(F)
80 NEXT F : B=(B+1) MOD 2 : GOTO 50
```

100  
130  
150  
170  
2474  
2077  
3976  
4950

TOTAL = 4950

### PROGRAMA 2

```
100 ' TESTE DE MOVIMENTO 2
110 SCREEN 1 : WIDTH 32 : COLOR 1,7,4
120 FOR F=0 TO 22
130 PRINT STRING$(32,"."); : NEXT F
140 A=VPEEK(8*ASC("."))
150 FOR F=8*ASC(".") TO 8*ASC(".") +6
160 VPOKE(F),VPEEK(F+1)
170 NEXT F
180 VPOKE F,A
190 GOTO 140
```

815  
107  
1029  
1901  
2403  
2753  
3776  
3906  
4080

TOTAL = 4080

### PROGRAMA 3

```
200 SCREEN1 : WIDTH 32
210 FOR F=48 TO 69
220 A$=A$+CHR$(F)
230 NEXT F
240 FOR F=1 TO 31
250 A=INT(1+RND(1)*21)
260 A$=RIGHT$(A$,A)+LEFT$(A$,21-A)
270 FOR G=1 TO 21:LOCATE F,G
280 PRINT MID$(A$,G,1):NEXT G
290 NEXT F
300 FOR F=48*8 TO 69*8+7
310 A=INT(8*RND(1))
```

330  
815  
100  
1497  
1007  
2883  
2507  
3876  
3829  
4594  
4007

320 IF RND(1)>.01 THEN VPOKE F,0 ELSE VP	8303
OKE F,2^A	
330 NEXT F	84F4
340 AZ=VPEEK(69*8+7)	8086
350 FOR FX=69*8+6 TO 48*8 STEP-1	7C03
360 VPOKE FX+1,VPEEK(FX)	8300
370 NEXT FX	848D
380 VPOKE FX+1,AZ	8879
390 GOTO 340	8AAA

1000	8AAA
------	------

#### PROGRAMA 4

400 SCREEN1 : WIDTH 32	2F9
410 FOR F=48 TO 69	795
420 A\$=A\$+CHR\$(F)	1067
430 NEXT F	785
440 FOR F=1 TO 31	1390
450 A=INT(1+RND(1)*21)	10F9
460 A\$=RIGHT\$(A\$,A)+LEFT\$(A\$,21-A)	208F
470 FOR G=1 TO 21:LOCATE F,G	2F69
480 PRINT MID\$(A\$,G,1):NEXT G	3849
490 NEXT F	3A8E
500 FOR F=48*8 TO 69*8+7	4504
510 A=INT(8*RND(1))	4C09
520 IF RND(1)>.01 THEN VPOKE F,0 ELSE VP	609C
OKE F,2^A	
530 NEXT F	6424
540 AZ=VPEEK(48*8)	6C54
550 FOR FX=48*8 TO 69*8+6	7894
560 VPOKE FX,VPEEK(FX+1)	8141
570 NEXT FX	827C
580 VPOKE FX,AZ	84FF
590 GOTO 540	867C

TOTAL	=	867C
-------	---	------



## PROGRAMA 5

600 ' PSEUDO-SCROLL A ESQUERDA	981
610 SCREEN 1 : WIDTH 32 : COLOR 1,7,4	0A8
620 X=B*ASC(" ")	12E8
630 B=0 : VPOKE X,&B00000001	14A5
640 VPOKE X,VPEEK(X)*2	209F
650 B=B+1 : IF B=7 THEN 630	2998
660 GOTO 640	2E42

TOTAL = 2E42

## PROGRAMA 6

670 ' PSEUDO-SCROLL A DIREITA	983
680 SCREEN 1 : WIDTH 32 : COLOR 1,7,4	0A8
690 X=B*ASC(" ")	1312
700 B=0 : VPOKE X,&B10000000	1380
710 VPOKE X,VPEEK(X)/2	218C
720 B=B+1 : IF B=7 THEN 700	2AFC
730 GOTO 710	2F00

TOTAL = 2F00

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulo 4.

## 2.9 - SCREEN 4

Quando comandamos SCREEN 1 ou SCREEN 2, o interpretador BASIC prepara a VRAM com as tabelas próprias para a tela selecionada e o VDP para agir sobre essas tabelas.

Analisando os procedimentos executados pelo micro quando usamos um comando SCREEN podemos idealizar uma forma de fazer a VRAM ser preparada com as tabelas da SCREEN 2 e as variáveis do sistema serem preparadas para operar sobre a SCREEN 1! Desse modo, poderemos conciliar a rapidez de operação da SCREEN 1 com todos os recursos de cores e formas da SCREEN 2.

A idéia é "enganar" o interpretador, fazendo-o pensar que está operando sobre a SCREEN 1, enquanto a VRAM e o VDP estarão preparados para a SCREEN 2. Note que as variáveis do sistema, preparadas pelo interpretador para uma dada SCREEN, funcionam como um "mecanismo inibitório" para o VDP, fazendo-o comportar-se sob rígido controle. Se usarmos diretamente as rotinas do BIOS, sem que o interpretador "saiba", poderemos evitar os "mecanismos inibitórios" e gerar algo parecido com uma "ESQUIZOFRENIA INDUZIDA" no micro.

Observe atentamente o programa a seguir. Digite-o e, antes de executá-lo, leia os comentários que o seguem.

### PROGRAMA SCREEN 4

```
100 COLOR 15,1,15
110 SCREEN 2
120 DEFINT A-Z
130 SCREEN 1
140 DEFUSR = &H7E
150 X = USR(0)
160 FOR F = 0 TO 2047
170   X = PEEK (&H18BF + F)
180   VPOKE (      F),X
190   VPOKE (&H800 + F),X
200   VPOKE (&H1000 + F),X
210 NEXT F
220 FOR F=0 TO 7
230 VPOKE 2048+255*8+F,255
240 VPOKE 4096+255*8+F,255
250 NEXT F
```

```
10F
33B
E2E
80F
B87
F28
148A
11FC
2615
20F4
36D2
393C
3EE2
4811
511F
5807
```

Antes de mais nada, lembre-se de como a VRAM fica dividida quando se usa a SCREEN 1 ou a SCREEN 2.

Uma vez estudada a estrutura da VRAM quando no modo SCREEN 1 ou SCREEN 2, podemos analisar o programa.

A linha 100 serve apenas para que a cor da borda da tela seja selecionada quando o comando SCREEN na linha 110 for executado.

A linha 110 seleciona a SCREEN 2 para que as tabelas da VRAM sejam preparadas para ela.

A linha 120 serve apenas para aumentar um pouco a velocidade de execução do programa, afinal ele necessita apenas de valores inteiros. Este recurso dos micros MSX é extremamente útil para esta finalidade.

A linha 130 seleciona a SCREEN 1 para que as variáveis do sistema indiquem ao interpretador o modo SCREEN 1. Note que até aqui não houve nenhum "truque" de programação digno de maiores explicações. O interpretador ainda não foi "enganado"! Isso só ocorrerá após a execução das próximas linhas.

As linhas 140 e 150 executam a rotina do BIOS SETGRP (em &H007E). Ela prepara o VDP para acessar as tabelas da SCREEN 2. Note que, com isso, já temos a VRAM preparada como SCREEN 1 (e residualmente como SCREEN 2) e o VDP preparado para acessar a SCREEN 2.

As linhas de 160 a 210 carregam as tabelas de caracteres da VRAM com os caracteres da ROM (de 0 a 255). Os desenhos dos 255 caracteres são definidos três vezes, uma para cada terço da tela. A tabela correspondente às linhas de 0 a 7 da tela ocupa os primeiros 2 Kbytes da VRAM. De 2 a 4 Kbytes está a tabela correspondente às linhas de 8 a 15. As linhas de 16 a 23 usam a tabela de caracteres entre 4 e 6 Kbytes.

As linhas de 220 a 250 apenas redefinem os desenhos do caractere 255 (cursor) para serem usados no segundo e terceiro terço da tela.

Note que a linha 180 poderia ter sido omitida sem nenhum problema, pois o comando SCREEN 1 da linha 130 já havia carregado os desenhos dos 255 caracteres no primeiro terço da tela.

## ALTERANDO AS CORES DOS CARACTERES

Na SCREEN 1 podemos alterar a cor de grupos de 8 caracteres. Por exemplo, se alterarmos a cor do caractere 3, as cores dos caracteres de 0 a 7 serão simultaneamente alteradas. Na SCREEN 2, podemos alterar a cor de cada um dos caracteres

individualmente. Podemos ainda ir mais além e definir, para cada caractere, 16 diferentes cores! Isso mesmo, 16 cores em cada caractere!

Acrescente ao programa anterior as linhas mostradas a seguir e rode-o novamente. Com isso, os caracteres dos números e dos parênteses terão suas cores redefinidas no 1º terço da tela!

```
260 X=8192+8*ASC("(")
270 FOR F=X TO X+7
280   VPOKE F,&B10000001
290 NEXT F
300 X=8192+8*ASC(")")
310 FOR F=X TO X+7
320   VPOKE F,&B10000001
330 NEXT F
340 X=8192+8*ASC("0")
350 Y=8192+8*ASC("9")+7
360 FOR F=X TO Y
370   VPOKE F,&B11010001
380 NEXT F
```

```
5F25
66E0
73F5
77B2
8055
8534
8905
8AB6
9084
9995
9DA4
A3BC
A5AD
```

100HL = A5A5

Agora vamos fazer com que o cursor seja redefinido com as 16 cores. Acrescente também ao programa as linhas mostradas a seguir. Depois, execute-o.

```
390 F=8192+8*255
400 VPOKE F+0,&B000001000
410 VPOKE F+1,&B000011001
420 VPOKE F+2,&B000101010
430 VPOKE F+3,&B000111011
440 VPOKE F+4,&B010011100
450 VPOKE F+5,&B010111101
460 VPOKE F+6,&B011011110
470 VPOKE F+7,&B011111111
480 F=10240+8*255
490 VPOKE F+0,&B000001000
500 VPOKE F+1,&B000011001
510 VPOKE F+2,&B000101010
520 VPOKE F+3,&B000111011
530 VPOKE F+4,&B010011100
540 VPOKE F+5,&B010111101
550 VPOKE F+6,&B011011110
560 VPOKE F+7,&B011111111
570 F=12288+8*255
580 VPOKE F+0,&B000001000
```

```
201E
27A5
27A3
27CC
27A4
2721
27D1
270C
2714
2711
2700
12C4
1A87
2172
27C6
27B3
3B13
4610
4CCA
5A66
```

```

590 VPOKE F+1,&B00011001
600 VPOKE F+2,&B00101010
610 VPOKE F+3,&B00111011
620 VPOKE F+4,&B01001100
630 VPOKE F+5,&B01011101
640 VPOKE F+6,&B01101110
650 VPOKE F+7,&B01111111

```

```

27EF
14CF
21B0
27DE
5C81
50F3
560E

```

TOTAL = 560E

Agora, o cursor está colorido!

Os exemplos apresentados são bem simples para facilitar a compreensão. Os recursos oferecidos pela SCREEN híbrida que apresentamos são, entretanto, muito mais vastos.

Você deve ter percebido que o programa demora vários segundos para ser executado. Isso é aceitável quando levamos em conta que o BASIC tem que acessar quase 16 Kbytes, entretanto para os programadores mais exigentes a demora pode ser um fator muito negativo.

Podemos pensar, então, em transformar o programa numa rotina em Linguagem de Máquina. Indo além, podemos pensar numa rotina que permita a implementação do comando SCREEN 4 no BASIC, de modo que ao ser executado ele gere a tela híbrida com a mesma velocidade que as outras SCREEN's. Para facilitar a alteração das cores dos caracteres podemos imaginar um novo comando do BASIC ou ainda o aproveitamento de comandos não implementados como o IPL ou o CMD. Isto, entretanto, já é assunto para um texto mais extenso.

Se você não quiser esperar, poderá encontrar estas e muitas outras idéias já executadas e analisadas no livro PROGRAMAÇÃO AVANÇADA EM MSX. Para estudar mais detalhadamente a estrutura da VRAM nas várias SCREEN's, as rotinas do BIOS e as Variáveis do Sistema, sugerimos a leitura atenta do livro APROFUNDANDO-SE NO MSX. Nesses dois livros os assuntos são tratados de forma bastante completa.

Exemplos e aplicações práticas comentadas passo a passo podem ser encontradas nos livros COLEÇÃO DE PROGRAMAS PARA MSX, volumes 1 e 2.

Para completar seu conhecimento sobre a SCREEN 1, veja a dica 2.Q (Arlequim Bêbado).

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulo 4.  
Programação Avançada em MSX - capítulo 2.

## 2.4 - SETORES COM CIRCLE

O comando CIRCLE do BASIC permite o traçado de arcos de circunferências e de perímetros de setores circulares.

Para traçar setores, basta usar valores negativos para os ângulos inicial e final. Veja o programa a seguir:

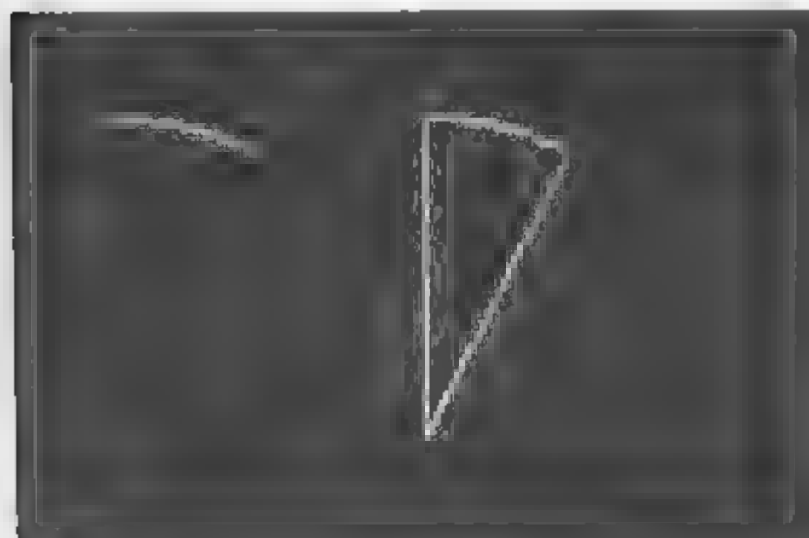
```
10 SCREEN 2
20 PI = 4*ATN(1)
30 CIRCLE (70,80),60,15,PI/3,PI/2,1
40 CIRCLE (130,80),60,15,-PI/3,-PI/2,1
50 GOTO 50
```

100  
200  
300  
400  
500  
600

A linha 20 define a variável PI, atribuindo-lhe o valor da constante matemática  $\pi$ .

A linha 30 traça um arco de circunferência normal.

A linha 40 traça o perímetro de um setor, pois os ângulos são negativos.



## 2.8 - REDEFININDO CARACTERES

Uma maneira rápida de redefinir caracteres é acessar a tabela de definição diretamente na VRAM. Essa tabela ocupa posições diferentes em cada tipo de tela (SCREEN's 0, 1, 2 ou 3). A seguir apresentamos 3 programas bem curtos que redefinem, apenas como exemplo, o formato do caractere de código 85 (ou 84H), originalmente a letra "A" nos SCREEN's 0, 1 e 2.

**PARA 4 SCREEN 8**

```

100 SCREEN 0
110 E=2048+8*65
120 FOR F=E TO E+7
130   READ A$
140   UPOKE F,VAL("&B"+A$)
150 NEXT F
160 PRINT : PRINT "CHR$(65)=";CHR$(65)
170 PRINT : LIST
180 DATA 00111100
190 DATA 01000010
200 DATA 10011001
210 DATA 10111101
220 DATA 10111101
230 DATA 10011001
240 DATA 01000010
250 DATA 00111100

```

總發行所：東京・丸の内區有樂町一丁目三番地

**PARA 8 SCREEN 1**

Apenas substitua as linhas 100 e 110 do programa anterior por:

```
100 SCREEN 1
110 F=B*65
```

**PAGE 1 SCREEN 2**

```

100 SCREEN 2
110 FOR F=6144 TO 6144+24*32-1
120   VPOKE F,65
130 NEXT F
140 FOR F=0 TO 4096 STEP 2048

```

150	FOR G=F+8*65 TO F+8*65+7	0034
160	READ A\$: VPOKE G,VAL("&B"+A\$)	0035
170	VPOKE G+8192,&B00011111	0036
180	NEXT G	0037
190	RESTORE	0038
200	NEXT F	0039
210	GOTO 210	0040
220	DATA 00111100	0041
230	DATA 01000010	0042
240	DATA 10011001	0043
250	DATA 10111101	0044
260	DATA 10111101	0045
270	DATA 10011001	0046
280	DATA 01000010	0047
290	DATA 00111100	0048

Essa forma de redefinir caracteres, entretanto, nem sempre é a melhor, pois quando um comando SCREEN é executado as tabelas da VRAM são automaticamente recarregadas. Para entender melhor esse problema rode novamente o programa feito para a SCREEN 0 e depois comando SCREEN 0. Você verá que o caractere de código 85 voltou a ser a letra "A". A redefinição feita na VRAM foi destruída pelo comando SCREEN.

Para contornar esse problema podemos redefinir toda a tabela de caracteres na própria RAM e fazer com que o comando SCREEN carregue essa nova tabela para a VRAM ao invés de usar a tabela da ROM. O programa apresentado a seguir se presta a facilitar a redefinição dos caracteres de forma mais permanente. Digite e grave o programa. Depois veremos como usá-lo.

1000	KEYOFF : SCREEN 0 : WIDTH 40	0049
1010	FOR E=&HB000 TO &HBAEF	0050
1020	READ C\$: POKE E,VAL("&H"+C\$)	0051
1030	NEXT E	0052
1040	DEFUSR=&HB000 : X=USR(0)	0053
1050	DATA 3E,01,21,AB,FC,77,CD,1E	0054
1060	DATA B1,CD,E5,B0,CD,26,B2,21	0055
1070	DATA 08,00,22,B7,FC,21,B8,00	0056
1080	DATA 22,B9,FC,21,CB,BA,7E,2F	0057
1090	DATA FE,00,28,06,CD,8D,00,23	0058
1100	DATA 18,F4,16,08,CD,57,B2,FE	0059
1110	DATA 53,20,06,AF,21,AB,FC,77	0060
1120	DATA C9,21,09,B0,E5,FE,43,CA	0061
1130	DATA 96,B2,FE,0D,2B,1F,0E,01	0062



1140 DATA FE,1C,28,11,0F,FF,FE,1D  
 1150 DATA 28,0B,0E,F0,FE,1F,28,05  
 1160 DATA 0E,10,FE,1F,C0,3A,C9,B2  
 1170 DATA 81,32,C9,B2,C9,CD,0F,B2  
 1180 DATA 16,02,CD,57,B2,FE,0D,CB  
 1190 DATA 21,65,80,E5,01,00,FE,FE  
 1200 DATA 20,28,24,0C,FE,4D,28,1F  
 1210 DATA FE,1C,28,11,0E,FF,FE,1D  
 1220 DATA 28,0B,0F,F8,FE,1E,28,05  
 1230 DATA 0E,0B,FF,1F,C0,3A,C9,B2  
 1240 DATA 81,F6,3F,32,4A,B2,C9,CD  
 1250 DATA 46,B2,3A,CA,B2,F5,0F,0F  
 1260 DATA 0F,E6,07,5F,16,00,FD,19  
 1270 DATA F1,F6,07,3F,CB,0B,CB,09  
 1280 DATA 3D,20,F9,FD,7E,00,A0,B1  
 1290 DATA FD,77,00,CD,E5,B0,CD,46  
 1300 DATA B2,CD,76,B2,CD,CB,B1,06  
 1310 DATA 0B,D5,F5,3E,0B,FD,5E,00  
 1320 DATA CD,EC,B1,E1,01,CD,E0,B1  
 1330 DATA FD,23,10,FD,C9,CD,46,B2  
 1340 DATA 0E,BF,1E,07,CD,C9,B1,06  
 1350 DATA 0B,0E,05,C5,D5,E5,06,0B  
 1360 DATA FD,7E,00,07,F5,9F,5F,3E  
 1370 DATA 05,CD,EC,B1,CD,D6,B1,CD  
 1380 DATA D6,B1,F1,10,EE,F1,D1,C1  
 1390 DATA CD,E0,B1,0D,20,DD,CD,E0  
 1400 DATA B1,FD,23,10,D4,C9,01,00  
 1410 DATA 0B,11,CB,B2,2A,20,F9,C5  
 1420 DATA D5,3A,1F,F9,CD,0C,00,FB  
 1430 DATA D1,C1,12,13,27,0B,7B,B1  
 1440 DATA 20,1D,CD,72,00,3A,E9,F3  
 1450 DATA 07,07,07,07,4F,3A,FA,F3  
 1460 DATA B1,01,00,1B,2A,C9,F1,CD  
 1470 DATA 56,00,11,0B,B1,01,0A,FF  
 1480 DATA 1E,06,11,11,CD,8A,B1,21  
 1490 DATA 06,11,01,8E,AA,1E,06,3E  
 1500 DATA 09,CD,8A,B1,21,30,31,01  
 1510 DATA 8F,FF,1E,06,3E,02,CD,8A  
 1520 DATA B1,AF,32,CA,B2,21,C9,B2  
 1530 DATA 77,E5,FD,C6,80,E1,34,20  
 1540 DATA F8,C9,F5,C5,E5,CD,CB,B1  
 1550 DATA C1,F1,5F,F1,F5,D5,E5,F5  
 1560 DATA C5,D5,E5,7B,CD,EC,B1,E1  
 1570 DATA D1,CD,E0,B1,0D,20,FA,C1  
 1580 DATA F1,3D,20,EB,E1,D1,F1,F5  
 1590 DATA C5,D5,F5,3E,01,CD,EC,B1  
 1600 DATA CD,E0,B1,10,F6,E1,D1,CD  
 1610 DATA D6,B1,0D,20,FA,C1,F1,3D  
 1620 DATA 20,E5,C9,06,00,50,CD,11

00000  
 00001  
 00002  
 00003  
 00004  
 00005  
 00006  
 00007  
 00008  
 00009  
 00010  
 00011  
 00012  
 00013  
 00014  
 00015  
 00016  
 00017  
 00018  
 00019  
 00020  
 00021  
 00022  
 00023  
 00024  
 00025  
 00026  
 00027  
 00028  
 00029  
 00030  
 00031  
 00032  
 00033  
 00034  
 00035  
 00036  
 00037  
 00038  
 00039  
 00040  
 00041  
 00042  
 00043  
 00044  
 00045  
 00046  
 00047  
 00048  
 00049  
 00050  
 00051  
 00052  
 00053  
 00054  
 00055  
 00056  
 00057  
 00058  
 00059  
 00060  
 00061  
 00062  
 00063  
 00064  
 00065  
 00066  
 00067  
 00068  
 00069  
 00070  
 00071  
 00072  
 00073  
 00074  
 00075  
 00076  
 00077  
 00078  
 00079  
 00080  
 00081  
 00082  
 00083  
 00084  
 00085  
 00086  
 00087  
 00088  
 00089  
 00090  
 00091  
 00092  
 00093  
 00094  
 00095  
 00096  
 00097  
 00098  
 00099  
 00100

1630 DATA 01,CD,14,01,57,C9,CB,0A  
 1640 DATA D0,C5,01,0B,00,09,C1,C9  
 1650 DATA 23,7D,E6,07,C0,C5,01,F8  
 1660 DATA 00,09,C1,C9,C5,47,CD,4A  
 1670 DATA 00,4F,7A,2F,A1,FB,03,30  
 1680 DATA 01,B2,05,28,0C,CB,0A,30  
 1690 DATA F0,CD,4D,00,CD,D9,B1,1B  
 1700 DATA E5,CD,4D,00,C1,C9,3A,CA  
 1710 DATA B2,F5,F6,07,07,4F,07,B1  
 1720 DATA C6,BF,4F,F1,E6,3B,01,5F  
 1730 DATA 0F,83,C6,07,5F,C9,3A,C9  
 1740 DATA B2,F5,CD,3C,02,16,0C,4F  
 1750 DATA F1,0F,0F,0F,01,1D,7C,B2  
 1760 DATA C6,08,5F,C9,E6,0F,57,07  
 1770 DATA 47,07,07,0B,02,C9,3A,C9  
 1780 DATA B2,6F,26,00,29,29,29,EB  
 1790 DATA FD,21,CB,02,FD,19,C9,06  
 1800 DATA 00,C5,D5,CD,7D,B2,D1,C1  
 1810 DATA 04,21,40,1F,1D,2C,00,20  
 1820 DATA 07,2B,7C,B5,20,F6,1B,E9  
 1830 DATA CB,40,C4,7B,B2,C3,9F,00  
 1840 DATA D5,CD,CB,B1,F1,47,5F,D5  
 1850 DATA E5,CD,4A,00,AA,1D,4D,00  
 1860 DATA CD,06,B1,1D,20,F3,F1,D1  
 1870 DATA CD,E0,B1,10,1A,C9,01,00  
 1880 DATA 08,11,80,BB,1D,1,20,F9  
 1890 DATA 21,CB,02,1D,B0,CD,3B,01  
 1900 DATA 07,07,F6,03,4F,06,00,21  
 1910 DATA C1,FC,09,CB,7E,2B,01,21  
 1920 DATA C5,FC,09,7E,07,07,07,07  
 1930 DATA E6,0C,B1,CB,FF,32,1F,F9  
 1940 DATA C9,00,00,01,24,79,1F,1F  
 1950 DATA 1F,E6,07,19,E5,2A,5E,EC  
 1960 DATA 7E,23,22,5E,EC,F1,4F,C9  
 1970 DATA AF,8E,2B,0D,79,BE,23,3E  
 1980 DATA 00,CB,CB,7E,23,2B,F8,1D  
 1990 DATA EF,23,7E,23,A1,8E,23,7E  
 2000 DATA 23,2B,0D,CB,7E,20,F1,7E  
 2010 DATA 07,23,30,F8,CB,7E,20,FA  
 2020 DATA CB,7E,CB,23,23,23,1B,F8  
 2030 DATA E5,C5,21,33,63,01,05,00  
 2040 DATA ED,01,C1,E1,2B,0D,C9,FE  
 2050 DATA E9,2B,0B,C9,FF,47,2B,03  
 2060 DATA FE,4D,C0,F5,3F,FF,32,47  
 2070 DATA EC,F1,C9,1B,C3,76,C9,E9  
 2080 DATA CD,D4,62,1B,0B,CD,D4,62  
 2090 DATA 47,CD,D4,62,CD,31,64,7B  
 2100 DATA CD,31,64,3E,40,C3,41,64  
 2110 DATA 79,E6,3B,1B,F3,3E,49,CB

0000  
 0001  
 0002  
 0003  
 0004  
 0005  
 0006  
 0007  
 0008  
 0009  
 0010  
 0011  
 0012  
 0013  
 0014  
 0015  
 0016  
 0017  
 0018  
 0019  
 0020  
 0021  
 0022  
 0023  
 0024  
 0025  
 0026  
 0027  
 0028  
 0029  
 0030  
 0031  
 0032  
 0033  
 0034  
 0035  
 0036  
 0037  
 0038  
 0039  
 0040  
 0041  
 0042  
 0043  
 0044  
 0045  
 0046  
 0047  
 0048  
 0049  
 0050  
 0051  
 0052  
 0053  
 0054  
 0055  
 0056  
 0057  
 0058  
 0059  
 0060  
 0061  
 0062  
 0063  
 0064  
 0065  
 0066  
 0067  
 0068  
 0069  
 0070  
 0071  
 0072  
 0073  
 0074  
 0075  
 0076  
 0077  
 0078  
 0079  
 0080  
 0081  
 0082  
 0083  
 0084  
 0085  
 0086  
 0087  
 0088  
 0089  
 0090  
 0091  
 0092  
 0093  
 0094  
 0095  
 0096  
 0097  
 0098  
 0099  
 0100  
 0101  
 0102  
 0103  
 0104  
 0105  
 0106  
 0107  
 0108  
 0109  
 0110  
 0111  
 0112  
 0113  
 0114  
 0115  
 0116  
 0117  
 0118  
 0119  
 0120  
 0121  
 0122  
 0123  
 0124  
 0125  
 0126  
 0127  
 0128  
 0129  
 0130  
 0131  
 0132  
 0133  
 0134  
 0135  
 0136  
 0137  
 0138  
 0139  
 0140  
 0141  
 0142  
 0143  
 0144  
 0145  
 0146  
 0147  
 0148  
 0149  
 0150  
 0151  
 0152  
 0153  
 0154  
 0155  
 0156  
 0157  
 0158  
 0159  
 0160  
 0161  
 0162  
 0163  
 0164  
 0165  
 0166  
 0167  
 0168  
 0169  
 0170  
 0171  
 0172  
 0173  
 0174  
 0175  
 0176  
 0177  
 0178  
 0179  
 0180  
 0181  
 0182  
 0183  
 0184  
 0185  
 0186  
 0187  
 0188  
 0189  
 0190  
 0191  
 0192  
 0193  
 0194  
 0195  
 0196  
 0197  
 0198  
 0199  
 0200  
 0201  
 0202  
 0203  
 0204  
 0205  
 0206  
 0207  
 0208  
 0209  
 0210  
 0211  
 0212  
 0213  
 0214  
 0215  
 0216  
 0217  
 0218  
 0219  
 0220  
 0221  
 0222  
 0223  
 0224  
 0225  
 0226  
 0227  
 0228  
 0229  
 0230  
 0231  
 0232  
 0233  
 0234  
 0235  
 0236  
 0237  
 0238  
 0239  
 0240  
 0241  
 0242  
 0243  
 0244  
 0245  
 0246  
 0247  
 0248  
 0249  
 0250  
 0251  
 0252  
 0253  
 0254  
 0255  
 0256  
 0257  
 0258  
 0259  
 0260  
 0261  
 0262  
 0263  
 0264  
 0265  
 0266  
 0267  
 0268  
 0269  
 0270  
 0271  
 0272  
 0273  
 0274  
 0275  
 0276  
 0277  
 0278  
 0279  
 0280  
 0281  
 0282  
 0283  
 0284  
 0285  
 0286  
 0287  
 0288  
 0289  
 0290  
 0291  
 0292  
 0293  
 0294  
 0295  
 0296  
 0297  
 0298  
 0299  
 0300  
 0301  
 0302  
 0303  
 0304  
 0305  
 0306  
 0307  
 0308  
 0309  
 0310  
 0311  
 0312  
 0313  
 0314  
 0315  
 0316  
 0317  
 0318  
 0319  
 0320  
 0321  
 0322  
 0323  
 0324  
 0325  
 0326  
 0327  
 0328  
 0329  
 0330  
 0331  
 0332  
 0333  
 0334  
 0335  
 0336  
 0337  
 0338  
 0339  
 0340  
 0341  
 0342  
 0343  
 0344  
 0345  
 0346  
 0347  
 0348  
 0349  
 0350  
 0351  
 0352  
 0353  
 0354  
 0355  
 0356  
 0357  
 0358  
 0359  
 0360  
 0361  
 0362  
 0363  
 0364  
 0365  
 0366  
 0367  
 0368  
 0369  
 0370  
 0371  
 0372  
 0373  
 0374  
 0375  
 0376  
 0377  
 0378  
 0379  
 0380  
 0381  
 0382  
 0383  
 0384  
 0385  
 0386  
 0387  
 0388  
 0389  
 0390  
 0391  
 0392  
 0393  
 0394  
 0395  
 0396  
 0397  
 0398  
 0399  
 0400  
 0401  
 0402  
 0403  
 0404  
 0405  
 0406  
 0407  
 0408  
 0409  
 0410  
 0411  
 0412  
 0413  
 0414  
 0415  
 0416  
 0417  
 0418  
 0419  
 0420  
 0421  
 0422  
 0423  
 0424  
 0425  
 0426  
 0427  
 0428  
 0429  
 0430  
 0431  
 0432  
 0433  
 0434  
 0435  
 0436  
 0437  
 0438  
 0439  
 0440  
 0441  
 0442  
 0443  
 0444  
 0445  
 0446  
 0447  
 0448  
 0449  
 0450  
 0451  
 0452  
 0453  
 0454  
 0455  
 0456  
 0457  
 0458  
 0459  
 0460  
 0461  
 0462  
 0463  
 0464  
 0465  
 0466  
 0467  
 0468  
 0469  
 0470  
 0471  
 0472  
 0473  
 0474  
 0475  
 0476  
 0477  
 0478  
 0479  
 0480  
 0481  
 0482  
 0483  
 0484  
 0485  
 0486  
 0487  
 0488  
 0489  
 0490  
 0491  
 0492  
 0493  
 0494  
 0495  
 0496  
 0497  
 0498  
 0499  
 0500  
 0501  
 0502  
 0503  
 0504  
 0505  
 0506  
 0507  
 0508  
 0509  
 0510  
 0511  
 0512  
 0513  
 0514  
 0515  
 0516  
 0517  
 0518  
 0519  
 0520  
 0521  
 0522  
 0523  
 0524  
 0525  
 0526  
 0527  
 0528  
 0529  
 0530  
 0531  
 0532  
 0533  
 0534  
 0535  
 0536  
 0537  
 0538  
 0539  
 0540  
 0541  
 0542  
 0543  
 0544  
 0545  
 0546  
 0547  
 0548  
 0549  
 0550  
 0551  
 0552  
 0553  
 0554  
 0555  
 0556  
 0557  
 0558  
 0559  
 0560  
 0561  
 0562  
 0563  
 0564  
 0565  
 0566  
 0567  
 0568  
 0569  
 0570  
 0571  
 0572  
 0573  
 0574  
 0575  
 0576  
 0577  
 0578  
 0579  
 0580  
 0581  
 0582  
 0583  
 0584  
 0585  
 0586  
 0587  
 0588  
 0589  
 0590  
 0591  
 0592  
 0593  
 0594  
 0595  
 0596  
 0597  
 0598  
 0599  
 0600  
 0601  
 0602  
 0603  
 0604  
 0605  
 0606  
 0607  
 0608  
 0609  
 0610  
 0611  
 0612  
 0613  
 0614  
 0615  
 0616  
 0617  
 0618  
 0619  
 0620  
 0621  
 0622  
 0623  
 0624  
 0625  
 0626  
 0627  
 0628  
 0629  
 0630  
 0631  
 0632  
 0633  
 0634  
 0635  
 0636  
 0637  
 0638  
 0639  
 0640  
 0641  
 0642  
 0643  
 0644  
 0645  
 0646  
 0647  
 0648  
 0649  
 0650  
 0651  
 0652  
 0653  
 0654  
 0655  
 0656  
 0657  
 0658  
 0659  
 0660  
 0661  
 0662  
 0663  
 0664  
 0665  
 0666  
 0667  
 0668  
 0669  
 0670  
 0671  
 0672  
 0673  
 0674  
 0675  
 0676  
 0677  
 0678  
 0679  
 0680  
 0681  
 0682  
 0683  
 0684  
 0685  
 0686  
 0687  
 0688  
 0689  
 0690  
 0691  
 0692  
 0693  
 0694  
 0695  
 0696  
 0697  
 0698  
 0699  
 0700  
 0701  
 0702  
 0703  
 0704  
 0705  
 0706  
 0707  
 0708  
 0709  
 0710  
 0711  
 0712  
 0713  
 0714  
 0715  
 0716  
 0717  
 0718  
 0719  
 0720  
 0721  
 0722  
 0723  
 0724  
 0725  
 0726  
 0727  
 0728  
 0729  
 0730  
 0731  
 0732  
 0733  
 0734  
 0735  
 0736  
 0737  
 0738  
 0739  
 0740  
 0741  
 0742  
 0743  
 0744  
 0745  
 0746  
 0747  
 0748  
 0749  
 0750  
 0751  
 0752  
 0753  
 0754  
 0755  
 0756  
 0757  
 0758  
 0759  
 0760  
 0761  
 0762  
 0763  
 0764  
 0765  
 0766  
 0767  
 0768  
 0769  
 0770  
 0771  
 0772  
 0773  
 0774  
 0775  
 0776  
 0777  
 0778  
 0779  
 0780  
 0781  
 0782  
 0783  
 0784  
 0785  
 0786  
 0787  
 0788  
 0789  
 0790  
 0791  
 0792  
 0793  
 0794  
 0795  
 0796  
 0797  
 0798  
 0799  
 0800  
 0801  
 0802  
 0803  
 0804  
 0805  
 0806  
 0807  
 0808  
 0809  
 0810  
 0811  
 0812  
 0813  
 0814  
 0815  
 0816  
 0817  
 0818  
 0819  
 0820  
 0821  
 0822  
 0823  
 0824  
 0825  
 0826  
 0827  
 0828  
 0829  
 0830  
 0831  
 0832  
 0833  
 0834  
 0835  
 0836  
 0837  
 0838  
 0839  
 0840  
 0841  
 0842  
 0843  
 0844  
 0845  
 0846  
 0847  
 0848  
 0849  
 0850  
 0851  
 0852  
 0853  
 0854  
 0855  
 0856  
 0857  
 0858  
 0859  
 0860  
 0861  
 0862  
 0863  
 0864  
 0865  
 0866  
 0867  
 0868  
 0869  
 0870  
 0871  
 0872  
 0873  
 0874  
 0875  
 0876  
 0877  
 0878  
 0879  
 0880  
 0881  
 0882  
 0883  
 0884  
 0885  
 0886  
 0887  
 0888  
 0889  
 0890  
 0891  
 0892  
 0893  
 0894  
 0895  
 0896  
 0897  
 0898  
 0899  
 0900  
 0901  
 0902  
 0903  
 0904  
 0905  
 0906  
 0907  
 0908  
 0909  
 0910  
 0911  
 0912  
 0913  
 0914  
 0915  
 0916  
 0917  
 0918  
 0919  
 0920  
 0921  
 0922  
 0923  
 0924  
 0925  
 0926  
 0927  
 0928  
 0929  
 0930  
 0931  
 0932  
 0933  
 0934  
 0935  
 0936  
 0937  
 0938  
 0939  
 0940  
 0941  
 0942  
 0943  
 0944  
 0945  
 0946  
 0947  
 0948  
 0949  
 0950  
 0951  
 0952  
 0953  
 0954  
 0955  
 0956  
 0957  
 0958  
 0959  
 0960  
 0961  
 0962  
 0963  
 0964  
 0965  
 0966  
 0967  
 0968  
 0969  
 0970  
 0971  
 0972  
 0973  
 0974  
 0975  
 0976  
 0977  
 0978  
 0979  
 0980  
 0981  
 0982  
 0983  
 0984  
 0985  
 0986  
 0987  
 0988  
 0989  
 0990  
 0991  
 0992  
 0993  
 0994  
 0995  
 0996  
 0997  
 0998  
 0999  
 1000  
 1001  
 1002  
 1003  
 1004  
 1005  
 1006  
 1007  
 1008  
 1009  
 1010  
 1011  
 1012  
 1013  
 1014  
 1015  
 1016  
 1017  
 1018  
 1019  
 1020  
 1021  
 1022  
 1023  
 1024  
 1025  
 1026  
 1027  
 1028  
 1029  
 1030  
 1031  
 1032  
 1033  
 1034  
 1035  
 1036  
 1037  
 1038  
 1039  
 1040  
 1041  
 1042  
 1043  
 1044  
 1045  
 1046  
 1047  
 1048  
 1049  
 1050  
 1051  
 1052  
 1053  
 1054  
 1055  
 1056  
 1057  
 1058  
 1059  
 1060  
 1061  
 1062  
 1063  
 1064  
 1065  
 1066  
 1067  
 1068  
 1069  
 1070  
 1071  
 1072  
 1073  
 1074  
 1075  
 1076  
 1077  
 1078  
 1079  
 1080  
 1081  
 1082  
 1083  
 1084  
 1085  
 1086  
 1087  
 1088  
 1089  
 1090  
 1091  
 1092  
 1093  
 1094  
 1095  
 1096  
 1097  
 1098  
 1099  
 1100  
 1101  
 1102  
 1103  
 1104  
 1105  
 1106  
 1107  
 1108  
 1109  
 1110  
 1111  
 1112  
 1113  
 1114  
 1115  
 1116  
 1117  
 1118  
 1119  
 1120  
 1121  
 1122  
 1123  
 1124  
 1125  
 1126  
 1127  
 1128  
 1129  
 1130  
 1131  
 1132  
 1133  
 1134  
 1135  
 1136  
 1137  
 1138  
 1139  
 1140  
 1141  
 1142  
 1143  
 1144  
 1145  
 1146  
 1147  
 1148  
 1149  
 1150  
 1151  
 1152  
 1153  
 1154  
 1155  
 1156  
 1157  
 1158  
 1159  
 1160  
 1161  
 1162  
 1163  
 1164  
 1165  
 1166  
 1167  
 1168  
 1169  
 1170  
 1171  
 1172  
 1173  
 1174  
 1175  
 1176  
 1177  
 1178  
 1179  
 1180  
 1181  
 1182  
 1183  
 1184  
 1185  
 1186  
 1187  
 1188  
 1189  
 1190  
 1191  
 1192  
 1193  
 1194  
 1195  
 1196  
 1197  
 1198  
 1199  
 1200  
 1201  
 1202



2610 DATA 27,44,41,C1,F3,44,C9,FB  
 2620 DATA 45,C9,D9,45,5B,D8,76,4B  
 2630 DATA 41,4C,D4,17,52,4C,C1,07  
 2640 DATA 52,4C,43,C1,1F,52,52,C1  
 2650 DATA 0F,52,52,43,C1,37,53,43  
 2660 DATA C6,00,FF,2A,26,FF,22,27  
 2670 DATA FF,3A,24,FF,32,25,FF,F9  
 2680 DATA 10,EF,02,11,EF,0A,12,C0  
 2690 DATA 40,09,C7,06,1F,CF,01,1F  
 2700 DATA 4C,C4,FF,CF,1D,FB,B8,0C  
 2710 DATA 41,44,C3,FF,C6,1D,FB,80  
 2720 DATA 0C,CF,09,0B,41,44,C4,FF  
 2730 DATA E6,1C,FB,A0,01,41,4E,C4  
 2740 DATA FF,CD,23,C7,C4,22,43,41  
 2750 DATA 4C,CC,FF,FE,1C,FB,B8,01  
 2760 DATA 43,D0,C7,05,06,CF,0B,04  
 2770 DATA 44,45,C3,FF,10,21,44,4A  
 2780 DATA 4E,DA,FF,E8,1B,FF,E3,19  
 2790 DATA FF,0B,0A,45,D8,FF,0B,1A  
 2800 DATA 49,LE,C7,04,06,CF,01,04  
 2810 DATA 49,4F,C3,C7,12,22,FF,C3  
 2820 DATA 21,1F,19,15,4A,D0,FF,1B  
 2830 DATA 21,17,20,20,4A,D2,FF,D3  
 2840 DATA 1B,4F,55,D4,FF,00,00,4E  
 2850 DATA 4F,D0,FF,F6,1C,FB,B8,01  
 2860 DATA 4F,D2,CF,C1,16,50,4F,D0  
 2870 DATA CF,C5,16,50,55,53,CB,FF  
 2880 DATA C9,00,C7,10,17,17,45,D4  
 2890 DATA C7,C7,0B,52,53,D4,FF,0F  
 2900 DATA 1D,FB,9B,01,53,42,C3,FF  
 2910 DATA D6,1C,FB,90,01,53,55,C2  
 2920 DATA FF,FF,1C,FB,AB,01,5B,4F  
 2930 DATA D7,00,74,66,6F,53,20,79  
 2940 DATA 61,69,54,00,C0,40,05,42  
 2950 DATA 49,D4,C0,00,05,52,45,D3  
 2960 DATA C0,C0,05,53,45,D4,FB,10  
 2970 DATA 01,52,C0,FB,1B,01,52,D2  
 2980 DATA FB,00,01,52,4C,C3,FB,0B  
 2990 DATA 01,52,52,C3,FB,20,01,53  
 3000 DATA 4C,C1,FB,2B,01,53,52,C1  
 3010 DATA FB,1B,01,53,52,C1,00,A9  
 3020 DATA 43,50,C4,B9,43,50,44,D2  
 3030 DATA A1,43,50,C9,B1,43,50,49  
 3040 DATA D2,AA,49,4E,C4,BA,49,4E  
 3050 DATA 44,02,A2,49,4E,C9,B2,49  
 3060 DATA 4E,49,D2,AB,4C,44,C4,BB  
 3070 DATA 4C,44,44,D2,A0,4C,44,C9  
 3080 DATA B0,4L,44,49,D2,44,4F,45  
 3090 DATA C7,BB,4F,54,44,D2,B3,4F

3000  
 3001  
 3002  
 3003  
 3004  
 3005  
 3006  
 3007  
 3008  
 3009  
 3010  
 3011  
 3012  
 3013  
 3014  
 3015  
 3016  
 3017  
 3018  
 3019  
 3020  
 3021  
 3022  
 3023  
 3024  
 3025  
 3026  
 3027  
 3028  
 3029  
 3030  
 3031  
 3032  
 3033  
 3034  
 3035  
 3036  
 3037  
 3038  
 3039  
 3040

3100 DATA 54,49,02,AB,4F,55,54,C4  
 3110 DATA A3,4F,55,54,C9,40,52,45  
 3120 DATA 54,C9,45,52,45,54,CE,6F  
 3130 DATA 52,4C,C4,67,52,52,C4,00  
 3140 DATA CF,4A,07,41,44,C3,FF,46  
 3150 DATA 0D,FF,56,0E,FF,FF,0F,49  
 3160 DATA CD,C7,40,02,49,CF,CF,4B  
 3170 DATA 2B,CF,43,29,F7,57,13,F7  
 3180 DATA 47,14,4C,C4,C7,41,03,4F  
 3190 DATA 55,D4,CF,42,07,57,42,C3  
 3200 DATA 00,2A,77,7E,71,96,22,9E  
 3210 DATA 09,86,19,8E,23,A6,29,AE  
 3220 DATA 34,86,35,8E,36,2B,39,E1  
 3230 DATA 46,E3,4E,E5,56,E4,5E,F9  
 3240 DATA 66,7E,70,71,72,71,74,75  
 3250 DATA 3F,8F,CD,F3,43,2B,07,FE  
 3260 DATA 31,2B,14,C3,47,43,21,0B  
 3270 DATA 67,CD,02,67,CD,9F,00,15  
 3280 DATA CD,84,42,F1,FF,03,1B,21  
 3290 DATA E1,67,7E,B7,CB,CD,86,42  
 3300 DATA 73,1B,17,45,44,49,54,4F  
 3310 DATA 52,70,41,4F,4D,4D,41,4E  
 3320 DATA 44,0D,0A,41,45,57,0D,4F  
 3330 DATA 49,73,54,20,5B,6E,31,5B  
 3340 DATA 2D,6E,72,5D,5B,0D,4C,4C  
 3350 DATA 49,57,54,20,5B,6E,31,5B  
 3360 DATA 2D,6E,72,5D,5B,0D,41,55  
 3370 DATA 54,4F,20,5B,6F,31,5B,21  
 3380 DATA 6E,72,5D,5B,0D,52,41,4F  
 3390 DATA 55,4D,70,5B,6E,31,5B,2C  
 3400 DATA 6E,72,5B,2C,6E,33,5D,5D  
 3410 DATA 5D,0D,44,45,4C,45,54,45  
 3420 DATA 20,6F,11,5D,2D,6E,72,5D  
 3430 DATA 0D,46,49,4F,44,73,0D,53  
 3440 DATA 45,41,52,43,4B,73,0D,4C  
 3450 DATA 53,45,41,52,43,4B,73,0D  
 3460 DATA 43,4B,41,4E,47,45,64,73  
 3470 DATA 31,64,5B,73,33,5D,0D,53  
 3480 DATA 41,56,45,20,22,66,69,6C  
 3490 DATA 65,20,6E,61,6D,65,22,0D  
 3500 DATA 4C,4F,41,44,20,22,66,69  
 3510 DATA 6C,65,20,6E,61,6D,65,22  
 3520 DATA 0D,4D,45,52,47,45,20,22  
 3530 DATA 66,69,6C,65,20,6E,61,6D  
 3540 DATA 65,22,0D,4D,41,50,0D,41  
 3550 DATA 5B,4E,55,50,4F,49,52,53  
 3560 DATA 44,4B,2F,7B,7B,5D,0D,42  
 3570 DATA 41,0D,00,0D,4D,4F,4E,49  
 3580 DATA 54,4F,52,20,43,4F,4D,4D

11304  
 11305  
 11306  
 11307  
 11308  
 11309  
 11310  
 11311  
 11312  
 11313  
 11314  
 11315  
 11316  
 11317  
 11318  
 11319  
 11320  
 11321  
 11322  
 11323  
 11324  
 11325  
 11326  
 11327  
 11328  
 11329  
 11330  
 11331  
 11332  
 11333  
 11334  
 11335  
 11336  
 11337  
 11338  
 11339  
 11340  
 11341  
 11342  
 11343  
 11344  
 11345  
 11346  
 11347  
 11348  
 11349  
 11350  
 11351  
 11352  
 11353  
 11354  
 11355  
 11356  
 11357  
 11358  
 11359  
 11360  
 11361  
 11362  
 11363  
 11364  
 11365  
 11366  
 11367  
 11368  
 11369  
 11370  
 11371  
 11372  
 11373  
 11374  
 11375  
 11376  
 11377  
 11378  
 11379  
 11380  
 11381  
 11382  
 11383  
 11384  
 11385  
 11386  
 11387  
 11388  
 11389  
 11390  
 11391  
 11392  
 11393  
 11394  
 11395  
 11396  
 11397  
 11398  
 11399  
 11400



4080 DATA 3A,C1,FC,BE,28,14,23,BE  
 4090 DATA 28,10,21,F0,69,11,00,C1  
 4100 DATA 01,C4,00,ED,80,CD,00,C1  
 4110 DATA 87,C9,FB,37,C9,21,C1,FC  
 4120 DATA 06,04,AF,E6,03,B6,C5,E5  
 4130 DATA 61,2E,10,F5,CD,0C,00,2F  
 4140 DATA 5F,F1,05,F5,CD,14,00,F1  
 4150 DATA D1,F5,D5,CD,0C,00,L1,47  
 4160 DATA 79,2F,5F,F1,F5,C5,LD,14  
 4170 DATA 00,C1,79,88,20,17,F1,2D  
 4180 DATA 20,D9,24,24,24,24,4F,7C  
 4190 DATA FE,40,28,03,FE,80,79,20  
 4200 DATA 08,79,E1,E1,C9,F1,E1,C1  
 4210 DATA 87,F2,EA,69,L6,04,FE,90  
 4220 DATA 38,84,23,3C,10,AD,37,C9  
 4230 DATA 3A,J0,FA,26,00,CD,1C,C1  
 4240 DATA 3A,J1,FA,26,40,CD,1C,C1  
 4250 DATA DB,AD,F5,3A,32,FA,D3,AB  
 4260 DATA 47,F1,FB,L9,CD,49,C1,FA  
 4270 DATA 29,L1,D8,AB,A1,80,D3,AB  
 4280 DATA C9,CD,9B,C1,28,13,E5,CD  
 4290 DATA 6E,C1,4F,06,00,7D,A4,B2  
 4300 DATA 21,L5,FC,09,77,E1,79,18  
 4310 DATA 08,1D,A1,C1,21,C5,FC,72  
 4320 DATA C9,F3,F5,7C,07,07,E6,03  
 4330 DATA 5F,1C,3E,C0,07,07,1D,20  
 4340 DATA FB,5F,2F,4F,F1,F5,E6,03  
 4350 DATA 47,04,3E,AB,C6,55,10,FC  
 4360 DATA 57,A1,47,F1,87,C9,F5,7A  
 4370 DATA F6,C0,4F,F1,F5,57,DB,AB  
 4380 DATA 47,E6,3E,B1,F5,7A,0F,0F  
 4390 DATA F6,03,57,14,3E,AB,L6,55  
 4400 DATA 15,20,FB,AJ,57,7B,2F,67  
 4410 DATA F1,CD,B4,C1,F1,E6,03,C9  
 4420 DATA 14,15,C0,47,7B,FE,03,7B  
 4430 DATA C9,0F,0F,E6,07,57,3A,FF  
 4440 DATA FF,2F,47,E6,FC,B2,57,J2  
 4450 DATA FF,FF,7B,C9,D3,AB,3A,FF  
 4460 DATA FF,2F,6F,A4,B2,32,FF,FF  
 4470 DATA 78,D3,AB,C9,00,2A,2A,20  
 4480 DATA 41,4F,4D,50,49,4C,41,44  
 4490 DATA 4F,52,20,41,53,4D,43,4F  
 4500 DATA 43,41,52,BA,8B,86,AB,B0  
 4510 DATA AD,BF,DF,BF,B3,BA,AF,B7  
 4520 DATA DF,D2,DF,D7,CF,CF,CE,D6  
 4530 DATA DF,C7,CB,CC,D2,CC,CD,CF  
 4540 DATA CD,FF,20,00,00,00,00,00

0000  
 0100  
 0200  
 0300  
 0400  
 0500  
 0600  
 0700  
 0800  
 0900  
 1000  
 1100  
 1200  
 1300  
 1400  
 1500  
 1600  
 1700  
 1800  
 1900  
 2000  
 2100  
 2200  
 2300  
 2400  
 2500  
 2600  
 2700  
 2800  
 2900  
 3000  
 3100  
 3200  
 3300  
 3400  
 3500  
 3600  
 3700  
 3800  
 3900  
 4000  
 4100  
 4200  
 4300  
 4400  
 4500  
 4600  
 4700  
 4800  
 4900  
 5000  
 5100  
 5200  
 5300  
 5400  
 5500  
 5600  
 5700  
 5800  
 5900  
 6000  
 6100  
 6200  
 6300  
 6400  
 6500  
 6600  
 6700  
 6800  
 6900  
 7000  
 7100  
 7200  
 7300  
 7400  
 7500  
 7600  
 7700  
 7800  
 7900  
 8000  
 8100  
 8200  
 8300  
 8400  
 8500  
 8600  
 8700  
 8800  
 8900  
 9000  
 9100  
 9200  
 9300  
 9400  
 9500  
 9600  
 9700  
 9800  
 9900

Com o programa digitado e gravado corretamente, comande RUN para executá-lo. A tela deverá estar como mostra a figura a seguir:



Você tem agora em seu micro um poderoso editor de caracteres com dois modos de operação: SELEÇÃO e EDIÇÃO. O modo SELEÇÃO permite a escolha do caractere a ser editado. O modo EDIÇÃO permite a alteração do seu "desenho". Logo após ser carregado o programa opera no modo SELEÇÃO. Experimente usar as teclas de setas e observe o que acontece com o cursor (na tabela de caracteres) e com o quadrado no canto superior direito do vídeo (CARACTERE AMPLIADO).

Para alterar ou redesenhar completamente um dado caractere, deve-se inicialmente levar o cursor até ele com a ajuda das teclas de setas.

Feito isso, deve-se entrar no modo de EDIÇÃO, pressionando a tecla RETURN. Assim procedendo, o cursor desaparecerá da tela e um pequeno ponto será visível no quadrado do CARACTERE AMPLIADO, onde poderá ser feita a edição.

Para apagar os pontos marcados do caractere a ser editado basta pressionar a BARRA DE ESPAÇOS.

Para marcar um ponto no caractere basta pressionar a tecla da letra "M" (de Marcar).



Após redesenhar o caractere, para voltar ao modo de SELEÇÃO, basta pressionar RETURN novamente.

Uma vez alterados ou redesenhados os caracteres, deve-se avisar ao programa que essa nova tabela deve ser usada. Para isso basta pressionar a tecla da letra "C" (de Confirmar).

Por fim, para sair do programa e retornar ao BASIC pressiona-se a tecla da letra "S" (de Sair).

Resumindo temos os seguintes comandos à nossa disposição:

- SETAS - Movem o cursor.
- - Confirma o uso da tabela redefinida;
- S - Sai do programa e retorna ao BASIC.
- RETURN - Passa do modo SELEÇÃO para o EDIÇÃO e vice-versa.
- ESPAÇO - Apaga pontos no caractere em edição;
- M - Marca pontos no caractere em edição.

Até agora vimos como usar o programa para gerar e assumir uma nova tabela de caracteres. Vamos aprender como usar essa nova tabela.

Após ter retornado ao BASIC é conveniente, antes de mais nada, salvar a nova tabela em fita ou em disco. Para isso, comande:

```
BSAVE "TABELA.DAT",&H0000,&H1300
```

A seguir, apague o programa em BASIC que está na memória do micro comandando NEW e SCREEN 1

Caso você queira carregar uma tabela, já salva em fita ou em disco e assumi-la, basta comandar:

```
BLOAD "TABELA.DAT"  
POKE &HF920,&H80  
POKE &HF921,&HBB
```

Só para o EXPERT,

```
POKE &HF91F,2
```

Só para o HOTBIT,

```
POKE &HF91F,3
```

A seguir, deve-se usar o comando SCREEN

Para voltar a operar com a tabela original da ROM, rode o programinha mostrado a seguir.

```

10 POKE &HF91F,0
20 POKE &HF920,&HBF
30 POKE &HF921,&H1B
40 SCREEN 1

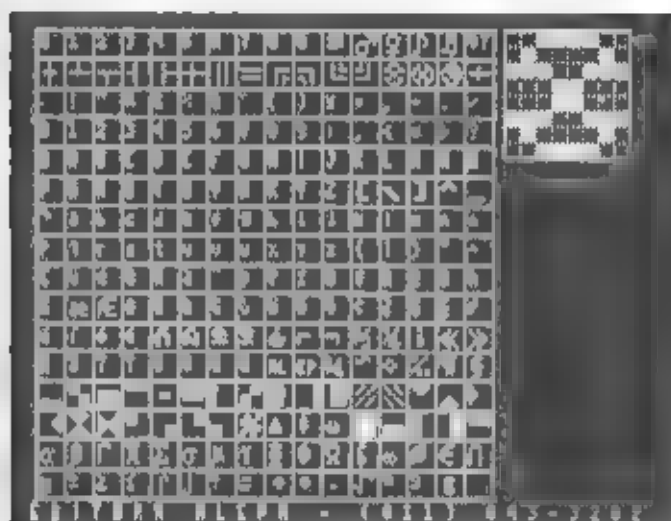
```

```

220
247
106
20F

```

Agora experimente redefinir a tabela de caracteres como mostrado abaixo:



#### BIBLIOGRAFIA RECOMENDADA:

Coleção de Programas para MSX v 1 - páginas 83 a 88  
 Coleção de Programas para MSX v 2 - páginas 10 a 24,  
 79 e 83 e 85  
 Aprofundando-se no MSX - capítulo 4  
 Programação Avançada em MSX - página 145.

## 2.C - SCROLL UP PARA A SCREEN 0

O programa apresentado a seguir gera uma rotina em L M para girar a SCREEN 0 para cima. A rotina em L M é carregada a partir de &HE000 e pode ser chamada pela instrução USR.

As linhas finais do programa em BASIC ilustram uma utilização da rotina em L.M. em conjunto com o BASIC.

100 REM -----	----	231-
110 REM SCROLL SCREEN 0 UP	Rubens Jr.	1005
120 REM -----	---	1006
130 FOR F &HE000 TO &HE046		2813
140 READ A\$: POKE F,VAL("&H"+A\$)		3554
150 NEXT F : DEFUSR0 &HE000		3010
160 DATA 21,00,00,22,44,E0,06,17		0011
170 DATA F3,21,20,00,ED,50,44,E0		0000
180 DATA 19,05,0D,2F,E0,C1,2A,44		0014
190 DATA E0,11,20,00,19,22,44,E0		0009
200 DATA 10,07,21,90,03,01,20,00		0002
210 DATA 31,20,0D,56,00,FB,C9,E5		0003
220 DATA D5,01,20,00,C5,11,10,FC		0002
230 DATA D5,0D,57,00,F1,C1,01,0D		0000
240 DATA 5C,00,E1,C9,00,00,00,00		0000
250 REM -	---	0000
260 REM	Exemplo de uso	0000
270 REM -----	-----	0000
280 SCREEN 0 : WIDTH 40 : KEY OFF		0000
290 X = 40 * RND(1)		0000
300 LOCATE X,20 : PRINT " . ";		0000
310 X = USR0(0)		0000
320 GOTO 290		0000

TOTAL = 000

## BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX capitulo 4  
 Programação Avançada em MSX capitulo 2.  
 Colecção de Programas para MSX v 2 páginas 130 e 131

## 2.D - SCROLL DOWN PARA A SCREEN 0

O programa apresentado a seguir gera uma rotina em L M para gerar a SCREEN 0 para baixo. A rotina em L M é carregada a partir de &HE000 e pode ser chamada pela instrução USR.

As linhas finais do programa em BASIC ilustram uma utilização da rotina em L M em conjunto com o BASIC

100 REM -----	0000
110 REM SCROLL SCREEN 0 DOWN-Rubens Jr.	0001
120 REM -----	0002
130 FOR F=&HE000 TO &HE049	0003
140 READ A\$: POKE F,VAL("&H"+A\$)	0004
150 NEXT F : DEFUSR0=&HE000	0005
160 DATA 21,70,03,22,47,F0,06,17	0006
170 DATA F3,21,28,00,ED,58,47,E0	0007
180 DATA 19,EB,C5,CD,32,E0,C1,2A	0008
190 DATA 47,E0,11,28,00,B7,ED,52	0009
200 DATA 22,47,E0,10,E4,21,00,00	0010
210 DATA 01,2B,00,3E,20,CD,56,00	0011
220 DATA F8,C9,E5,D5,01,28,00,C5	0012
230 DATA 11,18,FC,D5,CD,59,00,E1	0013
240 DATA C1,D1,LD,5C,00,E1,C9,00	0014
250 DATA 00,00,00,00,00,00,00,00	0015
260 REM -----	0016
270 REM Exemplo de uso	0017
280 REM -----	0018
290 SCREEN 0 : WIDTH 40 : KEY OFF	0019
300 X = 40 * RND(1)	0020
310 LOCATE X,0 : PRINT ". ";	0021
320 X = USR0(0)	0022
330 GOTO 300	0023

TOTAL = 1531

## BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX - capítulo 4  
Programação Avançada em MSX - capítulo 2.  
Colecção de Programas para MSX v 2 - páginas 130 e 131

## 2 E - SCROLL LEFT PARA A SCREEN 0

O programa apresentado a seguir gera uma rotina em L.M. para girar a SCREEN 0 para a esquerda. A rotina em L.M. é carregada a partir de &HE000 e pode ser chamada pela instrução USR.

As linhas finais do programa em BASIC ilustram uma utilização da rotina em L.M. em conjunto com o BASIC.

100 REM	-----	054
110 REM SCROLL SCREEN 0 LEFT-Rubens Jr.		0550
120 REM	-----	0550
130 FOR F=&HE000 TO &HE03F		0554
140 READ A\$: POKE F,VA("&H"+A\$)		0555
150 NEXT F: DEFUSR0=&HE000		055E
160 DATA 21,00,00,22,3D,E0,06,1B		0563
170 DATA F3,05,0D,1C,E0,C1,2A,3D		0604
180 DATA F0,11,20,00,19,22,3D,E0		0560
190 DATA 10,FF,FB,C9,15,01,20,00		81C7
200 DATA C7,11,1B,FC,D5,LD,59,00		87D0
210 DATA 21,19,FC,11,1B,FL,01,27		873A
220 DATA 00,ED,B0,3F,20,1C,11,C1		541D
230 DATA D1,CD,3C,00,C9,00,00,E1		608F
240 KIM	-----	0564
250 REM	Exemplo de uso	0565
260 REM	-----	0570
270 SCREEN 0: WIDTH 40: KEY OFF		0571
280 X = 23 * RND(1)		0C7A
290 LOCATE 37,X: PRINT ".":		200
300 X = USR0(0)		548
310 GOTO 280		693

TOTAL = 693

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX capítulo 4  
 Programação Avançada em MSX capítulo 2.  
 Coleção de Programas para MSX v.2 páginas 130 e 131

## 2 F - SCROLL RIGHT PARA A SCREEN 0

O programa apresentado a seguir gera uma rotina em L M para girar a SCREEN 0 para a direita. A rotina em L M é carregada a partir de &HE000 e pode ser chamada pela instrução USR

As linhas finais do programa em BASIC ilustram uma utilização da rotina em L M em conjunto com o BASIC

100 REM -	-----	65
110 R M SCROLL SCREEN 0 RIGHT-Rubens Jr		159
120 REM	-----	159
130 FOR F &HE000 TO &HLO7F		259
140 READ A\$ : POKE F,VA ("&H"+A\$)		359
150 NEXT F : DEF USR0=&HE000		459
160 DATA 21,00,00,22,30,F0,06,18		559
170 DATA F3,C5,CD,1C,E0,C1,2A,3D		659
180 DATA F0,11,2B,00,19,22,3D,E0		759
190 DATA 10,EF,FB,C9,E3,01,2B,00		859
200 DATA C5,11,18,FC,D5,1D,57,00		959
210 DATA 21,3E,FC,11,3E,FC,01,27		1059
220 DATA 00,ED,B0,3C,20,12,F1,C1		1159
230 DATA D1,CD,5C,00,C9,00,00,E1		1259
240 REM	-----	1359
250 REM	Exemplo de uso	1359
260 REM - - -		1459
270 SCREEN 0 : WIDTH 40 : KEY OFF		1559
280 X = 23 * RND(1)		1659
290 LOCATE 0,X : PRINT ".":		1759
300 X = USR0		1859
310 GOTO 280		1959

TOTAL = 600

## BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX - capítulo 4  
Programação Avançada em MSX - capítulo 2  
Coleção de Programas para MSX v 2 - páginas 130 e 131

## 2 G - SCROLL UP PARA A SCREEN 1

O programa apresentado a seguir gera uma rotina em L M para girar a SCREEN 1 para cima. A rotina em L M é carregada a partir de &HE000 e pode ser chamada pela instrução USR.

As linhas finais do programa em BASIC ilustram uma utilização de rotina em L M em conjunto com o BASIC.

Experimente usar esta dica em conjunto com as apresentadas no item 2 B (Movimentos na tela) você conseguirá na SCREEN 1 movimentos globais de SCROLL suaves e bastante rápidos.

```

100 REM
110 REM SCROLL SCREEN 1 UP - Rubens Jr.
120 REM
130 FOR F &HE000 TO &HE046
140 READ AS : POKE F,VAL("&H"+AS)
150 NEXT F : DEFUSR0=&HE000
160 DATA 21,00,1B,22,44,F0,06,17
170 DATA F3,21,20,00,E0,"B,44,F0
180 DATA 19,C5,CD,2F,E0,C1,2A,44
190 DATA E0,11,20,00,19,22,44,F0
200 DATA 10,E7,21,E0,1A,01,20,00
210 DATA 3E,20,00,"6,00,F0,C9,E5
220 DATA D5,01,20,00,C5,11,1B,FC
230 DATA D5,CD,59,00,E1,C1,01,CD
240 DATA 5C,00,E1,C9,00,00,C9,00
250 REM
260 REM      Exemplo de uso
270 REM
280 SCREEN 1 : WIDTH 32 : KEY OFF
290 X = 12 * RND(1)
300 LOCATE X,20 : PRINT ". ";
310 X = USR0(0)
320 GOTO 290

```

TOTAL = 105

## BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX capítulo 4  
 Programação Avançada em MSX capítulo 2.  
 Coleção de Programas para MSX v 2 página 12, 130 e 131

## 2.4 - SCROLL DOWN PARA A SCREEN 1

O programa apresentado a seguir gera uma rotina em L M para gerar a SCREEN 1 para baixo. A rotina em L M é carregada a partir de &HE000 e pode ser chamada pela instrução USR.

As linhas finais do programa em BASIC ilustram uma utilização da rotina em L M em conjunto com o BASIC.

Experimente usar esta dica em conjunto com as apresentadas no item 2.3 (Movimentos na tela). Você conseguirá na SCREEN 1 movimentos globais de SCROLL suaves e bastante rápidos!

100 REM	-----	-----	65E
110 REM "SCROLL SCREEN 1 DOWN Rubens Jr."			118A
120 REM	-----	-----	1F84
130 FOR I = &HE000 TO &HLE049			2884
140 READ A\$: POKE I, VAL("&H"+A\$)			3974
150 NEXT I : DEFUSR0 &HLE000			4074
160 DATA 21,00,1A,22,47,10,06,17			5080
170 DATA F3,21,20,00,ED,5B,47,E0			6174
180 DATA 19,EB,C5,CD,3,10,C1,2A			7674
190 DATA 47,E0,11,20,00,B7,FD,52			8270
200 DATA 22,47,E0,10,14,21,00,18			9774
210 DATA 01,20,00,3E,20,CD,56,00			AD05
220 DATA FB,C9,15,D5,01,20,00,17			56E7
230 DATA 11,1B,FC,D7,CD,59,00,F1			67E3
240 DATA C1,D1,CD,5C,00,E1,C9,00			78EE
250 DATA 00,0E,00,00,00,00,00,00			89D9
260 REM --	-----	-----	927E
270 REM		Exemplo de uso	AF28
280 REM	-----	-----	017E
290 SCREEN 1 : WIDTH 32 : KEY OFF			728
300 X = 40 * RND(1)			7E
310 LOCATE X,0 : PRINT "."			868
320 X = USR0(0)			934B
330 GOTO 300			1606

TOTAL = 1606

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulo 4  
 Programação Avançada em MSX - capítulo 2  
 Coletânea de Programas para MSX v 2 páginas 12 130 e 131.



## 2.1 - SCROLL LEFT PARA A SCREEN 1

O programa apresentado a seguir gera uma rotina em L M para girar a SCREEN 1 para a esquerda. A rotina em L M é carregada a partir de &HE000 e pode ser chamada pela instrução USR

As linhas finais do programa em BASIC ilustram uma utilização da rotina em L M em conjunto com o BASIC.

Experimente usar este dica em conjunto com as apresentadas no item 2.8 (Movimentos na tela). Você conseguirá na SCREEN 1 movimentos globais de SCROLL suaves e bastante rápidos!

100 REM	-----	65E
110 REM SCROLL SCREEN 1 LEFT-Rubens Jr.		6491
120 REM	-----	6491
130 FOR F=&HE000 TO &HE03F		20F5
140 READ A% : POKE F,VAL("&H"+A%)		3036
150 NEXT F : DEFUSR0 &HE000		903F
160 DATA 21,00,18,27,3D,F0,06,1B		4030
170 DATA F3,C7,CD,1C,E0,C1,2A,3D		606E
180 DATA E0,11,20,00,19,2F,3D,E0		7052
190 DATA 10,EF,F8,C9,E9,01,20,00		804E
200 DATA C5,11,18,FC,D5,CD,59,00		904A
210 DATA 21,19,FC,11,18,FC,01,1F		900B
220 DATA 00,ED,B0,3E,20,12,F1,C1		9001
230 DATA D1,CD,5C,00,C9,00,00,E1		A000
240 REM	-----	B0F0
250 REM	Exemplo de uso	C080
260 REM	-----	EC08
270 SCREEN 1 : WIDTH 32 : KEY OFF		F000
280 X = 23 * RND(1)		F000
290 LOCATE 31,X : PRINT ".";		4000
300 X = USR0()		4000
310 GOTO 280		6000

TOTAL = 60E

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX capítulo 4.  
 Programação Avançada em MSX capítulo 2  
 Coleção de Programas para MSX v 2 - páginas 12, 130 e 131

## 2.2 - SCROLL RIGHT PARA A SCREEN 1

O programa apresentado a seguir gera uma rotina em L M para girar a SCREEN 1 para a direita. A rotina em L M é carregada a partir de &HE000 e pode ser chamada pela instrução USR.

As linhas finais do programa em BASIC ilustram uma utilização da rotina em L M, em conjunto com o BASIC.

Experimente usar esta dica em conjunto com as apresentadas no item 2.8 (Movimentos na tela) - você conseguirá na SCREEN 1 movimentos globais de SCROLL suaves e bastante rápidos!

```

100 REM -----
110 REM SCROLL SCREEN 1 RIGHT Rubens Jr
120 REM
130 FOR F=&HE000 TO &H103F
140   READ A$: POKE F,VAL("&H"+A$)
150 NEXT F : DEF SR0=&HE000
160 DATA 21,00,1B,22,3D,E0,06,1B
170 DATA F3,C5,CD,1C,E0,C1,2A,3D
180 DATA E0,11,20,00,19,27,3D,E0
190 DATA 10,EF,FB,C9,E5,01,20,00
200 DATA C5,11,1B,FC,D5,CD,59,00
210 DATA 21,36,FC,11,37,FC,01,1F
220 DATA 00,ED,BB,3E,20,12,E1,C1
230 DATA D1,CD,5C,00,C9,00,00,E1
240 REM
250 REM      Exemplo de uso
260 REM
270 SCREEN 1 : WIDTH 32 : KEY OFF
280 X = 23 * RND(1)
290 LOCATE 0,X : PRINT ". ";
300 X = LSR0(0)
310 GOTO 280

```

TOTAL = 574

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulo 4.  
 Programação Avançada em MSX - capítulo 2  
 Coleção de Programas para MSX v 2 - páginas 12, 130 e 131

## 2.2 - CENTRALIZANDO CARACTERES

Os caracteres são desenhados dentro de uma matriz de 8x8 posições. Muitos deles usam apenas o lado esquerdo dessa matriz e há casos em que ao serem impressos na SCREEN 1 ou SCREEN 2 ficam fora de alinhamento. Podemos evitar isso de uma forma bem simples e rápida redefinindo os caracteres. A título de exemplo vamos 'centralizar' os caracteres dos números e das letras maiúsculas na SCREEN 1.

```

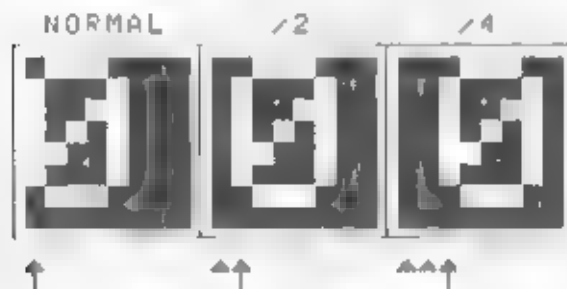
100 SCREEN 1
110 FOR F=6144 TO 6144+24*32-1 STEP 2
120   VPOKE F,ASC(" ")
130 NEXT F : LOCATE 10,10
140 PRINT "0123456789" : LOCATE 2,12
150 PRINT "ABCDEFGHIJKLMN OPQRST UVWXYZ"
160 FOR F=ASC("0") TO 8*ASC("9")+7
170   VPOKE F,VPEEK(F)/2
180 NEXT F
190 FOR F=ASC("A") TO 8*ASC("Z")+7
200   VPOKE F,VPEEK(F)/2
210 NEXT F

```

```

123456789
ABCDEFGHIJKLMN
OPQRST UVWXYZ

```



INTELLIGENTIA BICENTENARIA

Coleção de Programas para MSX v 2 - páginas 72 e 73.  
Aprofundando-se no MSX - página 90

## 2.1 - ANIMAÇÃO COM SPRITES

Se você definir vários SPRITE's com uma figura em posições sucessivas e os colocar na tela sempre na mesma camada, em sequência, eles darão o efeito de animação.

Se você quiser alterar o desenho do "boneco" no programa listado a seguir, uma boa idéia é eliminar a linha 98 digitando seu conteúdo manualmente no modo direto. A seguir você interrompe a listagem na sequência de linhas DATA que definem o SPRITE a ser alterado, modifica o desenho e comanda RUN. Desta forma você pode congelar o boneco com STOP e alterar os 0's e 1's da linha DATA, visualizando o efeito anterior à alteração.

Obviamente você pode definir SPRITE's de 16x16 pixels, obtendo figuras muito mais detalhadas.

Esse programa é apenas um exemplo (parece um bêbado tentando dançar BREAK!) mas pode ser alterado e melhorado conforme sua imaginação.

Pegue papel quadriculado e lápis e monte sua história!

```
90 SCREEN 1
100 GOSUB 220:SPRITE$(0)=S$
110 GOSUB 220:SPRITE$(1)=S$
120 GOSUB 220:SPRITE$(2)=S$
130 GOSUB 220:SPRITE$(3)=S$
140 GOSUB 220:SPRITE$(4)=S$
150 FOR I=1 TO 250 STEP 10
160 FOR S=0 TO 4
170 PUT SPRITE 0,(I+S*2,30),B,S
180 FORT=0TO50:NEXTT
190 NEXT S
200 NEXT I
210 GOTO 150
220 S$=""
230 FOR C=1 TO 8
240 READ K$
250 S$ S$+CHR$(VAL("&B"+K$))
260 NEXT C
270 RETURN
280 DATA 00011000
290 DATA 00011000
```

```
111-
66E
587
0102-
0173-
0133-
2687-
21E2
5787
0030
3-88
4212
4333
4922
4F1D
426F
511F
6411
6681
7048
7810
```

```

300 DATA 00111100
310 DATA 00111100
320 DATA 00111100
330 DATA 00011000
340 DATA 00011000
350 DATA 00011100
360 REM-----
370 DATA 00011000
380 DATA 00011000
390 DATA 00111100
400 DATA 01011010
410 DATA 01011010
420 DATA 00010100
430 DATA 00100110
440 DATA 00110000
450 REM---
460 DATA 00011000
470 DATA 00011000
480 DATA 00111100
490 DATA 01011010
500 DATA 10011001
510 DATA 00100100
520 DATA 01000010
530 DATA 01100011
540 REM-----
550 DATA 00011000
560 DATA 00011000
570 DATA 00111100
580 DATA 01011010
590 DATA 10011010
600 DATA 01100100
610 DATA 01000100
620 DATA 00000110
630 REM
640 DATA 00011000
650 DATA 00011000
660 DATA 00111100
670 DATA 00111100
680 DATA 01011010
690 DATA 00101000
700 DATA 00101100
710 DATA 00111110
720 REM-----

```

```

3010
3020
3030
3040
3050
3060
3070
3080
3090
3100
3110
3120
3130
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260
3270
3280
3290
3300
3310
3320
3330
3340
3350
3360
3370
3380
3390
3400
3410
3420
3430
3440
3450
3460
3470
3480
3490
3500
3510
3520
3530
3540
3550
3560
3570
3580
3590
3600
3610
3620
3630
3640
3650
3660
3670
3680
3690
3700
3710
3720

```

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX capítulo 4  
 Programação Avançada em MSX - capítulo 2.



## 2.M - LETRAS AMPLIADAS

O programa apresentado a seguir gera uma tela padrão na SCREEN 2 onde podem ser inseridas 4 mensagens. Digite-o e execute-o. Depois, experimente alterar o conteúdo das linhas 135, 140, 145 e 150. Experimente também alterar o caractere entre as aspas na linha 130. A tela gerada será gravada em disco ou em fita pela linha 705. Se desejar, altere o nome do arquivo

100 COLOR 15,1,1	0100
105 SCREEN 2	0105
110 OPEN"GRP:" AS #1	0110
115 REM	0115
120 REM define mensagens	0120
125 REM -----	0125
130 X\$(0)="0"	0130
135 X\$(1)="XXXXX"	0135
140 X\$(2)="XXXXXXXXXXXXX"	0140
145 X\$(3)="XXXXXXXXXXXXX"	0145
150 X\$(4)="XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"	0150
155 REM	0155
160 REM desenha mensagem	0160
165 REM -----	0165
170 X=6.5+(245-8*LEN(X\$(4)))/2	0170
175 PSET(X,170),POINT(X,170)	0175
180 PRINT #1, X\$(4)	0180
185 PSET(X+1,170),POINT(X+1,170)	0185
190 PRINT #1, X\$(4)	0190
195 REM	0195
200 REM desenha linhas horizontais	0200
205 REM	0205
210 COLOR 9	0210
215 YY 1	0215
220 FOR F=6 TO 191 STEP .2	0220
225 F = F + YY	0225
230 LINE(8,F)-(247,F),9	0230
235 YY = YY + .4	0235
240 NEXT F	0240
245 REM	0245
250 REM desenha a moldura	0250
255 REM -----	0255
260 FOR F=0 TO 255 STEP 8	0260
265 PSET(F,0),4	0265
270 PRINT #1,X\$(0)	0270
275 PSET(F,191-7),4	0275
280 PRINT #1,X\$(0)	0280
285 NEXT F	0285

290	FOR F=0 TO 191 STEP 8	8020
295	PSET(0,F),4	8021
300	PRINT #1,X\$(0)	8022
305	PSET(255-7,F),4	8023
310	PRINT #1,X\$(0)	8024
315	NEXT F	8025
320	REM	8026
325	REM desenha texto	8027
330	REM -----	8028
335	COLOR 14	8029
340	EN=PEEK(&HF920)+256*PEEK(&HF921)	8030
345	FOR TX=1 TO 3	8031
350	A\$=X\$(TX)	8032
355	Q=0	8033
360	FOR F=1 TO LEN(A\$)	8034
365	A=EN+8*ASC(MID\$(A\$,F,1))	8035
370	FOR G=0 TO 7	8036
375	B\$=BIN\$(PEEK(A+G))	8037
380	B\$=RIGHT\$("00000000"+B\$,8)	8038
385	FOR H=1 TO 8	8039
390	IF MID\$(B\$,H,1)="1" THEN Q=Q+1	8040
395	NEXT H	8041
400	NEXT G	8042
405	NEXT F	8043
410	P=2*Q-1	8044
415	DIM X(P),Y(P),S(P),T(P)	8045
420	E=0	8046
425	I=5	8047
430	FOR F=1 TO LEN(A\$)	8048
435	A=EN+8*ASC(MID\$(A\$,F,1))	8049
440	FOR G=0 TO 7	8050
445	B\$=BIN\$(PEEK(A+G))	8051
450	B\$=RIGHT\$("00000000"+B\$,8)	8052
455	FOR H=1 TO 8	8053
460	IF MID\$(B\$,H,1)="0" THEN 490	8054
465	X(E)=I+H-1	8055
470	Y(E)=B-G	8056
475	X(E+Q)=X(E)	8057
480	Y(E+Q)=Y(E)	8058
485	E=E+1	8059
490	NEXT H	8060
495	NEXT G	8061
500	REM	8062
505	REM passo horizontal	8063
510	REM -----	8064
515	PS=6	8065
520	I=I + PS	8066
525	NEXT F	8067
530	REM	8068

535	REM parâmetro de escala X	8131
540	REM -----	8132
545	PX=3	8133
550	IF TX=1 THEN PX=5	8134
555	REM	8135
560	REM parâmetro de escala Y	8136
565	REM -----	8137
570	PY=5	8138
575	IF TX=1 THEN PY=7	8139
580	REM	8140
585	REM acha posições na tela	8141
590	REM -----	8142
595	XI=INT((230-LEN(A\$)*PS*PX)/2)	8143
600	IF TX=1 THEN XI=XI-8	8144
605	YI=30+45*TX	8145
610	IF TX=1 THEN YI=YI-1	8146
615	IF TX=2 THEN YI=YI-1	8147
620	IF TX=3 THEN YI=YI+5	8148
625	DEFFN A(F)=XI+PX*X(F)	8149
630	DEFFN B(F)=YI-PY*Y(F)	8150
635	REM	8151
640	REM "plota" círculos das letras	8152
645	REM -----	8153
650	FOR F=0 TO P-1	8154
655	CX=FN A(F) :REM x do centro	8155
660	CY=FN B(F) :REM y do centro	8156
665	AX=1.7 :REM achatamento	8157
670	RD=4 :REM raio	8158
675	IF TX=1 THEN AX=1.4 : RD=6	8159
680	CIRCLE(CX,CY),RD,,,AX	8160
685	PAINT(CX,CY)	8161
690	NEXT F	8162
695	ERASE X,Y,S,T	8163
700	NEXT TX	8164
705	BSAVE"TELAXXXX.SCR",0,&H3FFF,S	8165
710	GOTO 710	8166

MOQCL = 2434

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX capítulo 4.  
Programação Avançada em MSX - capítulo 2



## 2.2 - ENTENDENDO O COMANDO DRAW

O comando DRAW do BASIC MSX é, na realidade, uma macro-linguagem gráfica que permite a confecção de desenhos nas telas 2 e 3.

Sua sintaxe é simples:

DRAW <expressão string>

A montagem do desenho é feita através da expressão string segundo regras bem determinadas e de fácil memorização. Para conhecê-las basta consultar o verbete DRAW do dicionário de comandos do livro LINGUAGEM BASIC MSX.

A melhor maneira de se familiarizar com esse comando é gerar uma variável string contendo os comandos desejados e ver o efeito na tela.

Para facilitar a visualização do desenho obtido é conveniente "reticular" a tela (de 10 em 10 pontos, por exemplo) antes da execução do desenho. Experimente digitar o programa a seguir:

```
10 SCREEN 2
20 FOR L=0 TO 191 STEP 10
30   FOR C=0 TO 255 STEP 10
40     PSET (C,L)
50   NEXT C
60 NEXT L
70 A$="BM60,60U10F10D10L5U5L5D5L10"
80 DRAW A$
90 GOTO 90
```

0000  
684  
042  
142  
1104  
1444  
1044  
1444  
2111

0000 - 0011

Agora vá alterando a variável A\$ definida na linha 80 com novos sub-comandos do DRAW e veja os efeitos gerados por suas experiências até se familiarizar com esse poderoso comando.

## BIBLIOGRAFIA RECOMENDADA.

Linguagem Basic MSX    páginas 56 a 58  
Coleção de Programas para MSX v 1 - página 54  
Coleção de Programas para MSX v 2    página 52  
Curso de BASIC v 1    páginas 61 e 62

## 2.0 - "WARP 6" NA SCREEN 2

Você está indo para o planeta da Princesa Vespa para salvá-la do terrível "Capacete Preto"! Para chegar à você passa por um aglomerado de estrelas com velocidade hiperfônica. Como simular este efeito? Digite o programa a seguir e boa viagem!

```

100 SCREEN 2:DEFINT I-S,X-Y
110 SPRITE$(1)=CHR$(0)+CHR$(0)+CHR$(0)+CHR$(0)+CHR$(0)+CHR$(8)+CHR$(0)+CHR$(0)+CHR$(0)
120 DIM X(31),Y(31),C(31),D(31),VX(31),VY(31),X1(31),Y1(31)
130 FOR I=0 TO 31
140 X(I)=80+INT(B0*RND(TIME))
150 FOR I=0 TO INT(300*RND(3)):NEXTT
160 Y(I)=40+INT(B0*RND(TIME))
170 IF X(I)=125 THEN GOTO 140
180 X(0)=126:Y(0)=86
190 PUT SPRITE I,(X(I),Y(I)),15,1
200 D(I)=(Y(I)-85)/(X(I)-125)
210 NEXT I
220 FOR I=1 TO 1000
230 FOR I=0 TO 31
240 VX(I)=SGN(X(I)-125)*INT^3/1000
250 VY(I)=SGN(Y(I)-85)*ABS(VX(I)*D(I))
260 X1(I)=X(I)+VX(I):Y1(I)=Y(I)+VY(I)
270 ON ERROR GOTO 330
280 IF X1(I)<0 OR X1(I)>225 OR Y1(I)<0 OR Y1(I)>191 THEN C=14
290 PLT SPRITE I,(X1(I),Y1(I)),15,C,1
300 C=0
310 NEXT I
320 NEXT T
330 GOTO 330

```

10026 - 628

## BIBLIOGRAFIA RECOMENDADA:

Apofundando-se no MSX - capítulo 4.

## 2 P - "SPRITEANDO" A TABELA DE CARACTERES

Muitas vezes sentimos a necessidade de fazer letras ou símbolos se movimentarem na tela como se fossem SPRITES. Isso é simples de ser feito, como podemos definir 256 SPRITES (8x8) e o MSX dispõe de 256 caracteres podemos transformar cada caractere num SPRITE!

O programa a seguir procura o endereço do começo da tabela de caracteres na variável EN (contida nos endereços &HF920 e &HF921) e transfere sua configuração para a região da VRAM reservada para os desenhos dos SPRITES.

```
10 SCREEN 1 : DEFINIT A Z : KEY OFF
20 EN = PEEK(&HF920)+256*PEEK(&HF921)
30 FOR F = 0 TO 2047
40 VPOKE 14336+F, PEEK(EN+F)
50 NEXT
```

64  
1038  
1414  
1442  
1028

Após rodar este programa cada SPRITE corresponderá a um dos 256 caracteres do MSX sendo seu número o próprio código ASC do caractere.

Para ver um dos muitos efeitos possíveis a serem empregados com esse recurso digite a complementação do programa, a seguir.

```
60 INPUT A$
70 L=LEN(A$) : C=3
80 FOR I=1 TO L
90 CH$=MID$(A$,I,1)
100 CH=ASC(CH$)
110 FOR Y=0 TO 96
120 PUT SPRITE I,((C+I)*8,Y-1),,CH
130 NEXT
140 LOCATE C+I-2,12,0 : PRINT CH$
150 NEXT
```

2626  
2648  
2AC4  
3111  
3879  
3F14  
4F59  
5073  
5F68  
601F

Para rodar o programa se você já rodou o programa anterior basta digitar GOTO 60. Desta forma não perdamos o tempo de transferência da tabela de caracteres para a área de SPRITES da VRAM. O programa pede a entrada de uma string (seu nome por exemplo). Evite os caracteres gráficos de código entre 0 e 31.

Implemente agora seu programa de maneira a aceitar qualquer caractere.

## 2.0 - ARLEQUIM BÊBADO

Quando at vamos a SCREEN 1, os 32 bytes da VRAM compreendidos entre os endereços 8192 e 8223 ficam reservados para atributos de cores. Cada um desses bytes define a cor de frente e a cor de fundo de um conjunto de 8 caracteres (8\*32=256).

Digite o programa a seguir para entender melhor este mecanismo.

100 SCREEN 1:KEY OFF:DEFINT A Z	619
110 FOR L=0 TO 15	61E
120 FOR C=0 TO 15	634
130 VPOKE 8192+32*L+C,16*L+C	1718
140 NEXT C	1928
150 NEXT L	1970
160 LOCATE 3,20,0:PRINT"BYTE ALTERADO=";	23AD
170 GOTO 190	2710
180 A STICK(0):IF A=0 THEN GOTO 180	3203
190 I I=-(A/3)+(A/7) 2*(A/5)+2*(A/1)	45F4
200 COLOR 15,1,1	4912
210 VPOKE 8192+I,&B10111000	5712
220 LOCATE 17,20:PRINT 8192+I	6439
230 IF STICK(0)<>0 THEN GOTO 230	6701
240 GOTO 180	6921

A linha 100 configura o VDP para SCREEN 1, apaga as telas de função e define todas as variáveis como inteiras para tornar o programa mais rápido.

As linhas de 110 a 150 colocam todos os caracteres na tela através do VPOKE para evitar problemas na impressão dos caracteres de controle (0 a 31 e 127) através da instrução PRINT.

A linha 210 insere num dos 32 bytes citados um valor que define a cor de frente e a cor de fundo segundo o seguinte critério: os 4 bits da esquerda definem a cor de frente (no nosso exemplo &B1011 = 11 = amarelo) e os 4 bits da direita a cor de fundo (&B1000 = 8 = vermelho).

Se você quiser "vpokear" um número em decimal basta calculá-lo segundo a regra

**VALOR DECIMAL=(COR DE FRENTE)\*16+(COR DE FUNDO)**

No nosso exemplo o valor decimal seria,

$$11*16 + 8 = 184$$

Experimente substituir esse valor na linha 210 e

depois invente outras combinações de cores

A linha 190 permite alterar o endereço do VPOKE usando o recurso do "parênteses lógico"

Se a afirmação for verdadeira seu valor será -1 se for falsa, será 0. Dessa forma, pressionando as teclas de setas ( STICK(0) ) você pode alterar o valor de I e portanto, o endereço do byte da VRAM a ser alterado

Assim por meio desse programa você pode visualizar quais são os grupos de 8 caracteres que têm sua cor alterada em função do byte de atributo que você mudou.

Se você, por exemplo, alterar os bytes B197, B198 e B199 com os comandos

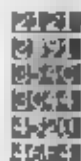
```
VPOKEB197,1B4:VPOKEB198,1B4:VPOKEB199,1B4
```

e comandar LIST para um programa em BASIC na SCREEN 1, verá uma listagem colorida apenas nos algarismos e símbolos aritméticos.

Note que a alteração no byte que inclui o caractere "espaço vazio" colore toda a tela ao redor da tabela de caracteres (pois ela está cheia de espaços vazios) e que uma alteração no último dos 32 bytes de atributos de cor (B233) altera a cor do cursor (caractere de código 255) permitindo uma visualização mais fácil do mesmo na hora de editar um programa

Apague as linhas de 160 a 240 com "DELETE 160-240" e acrescente estas linhas ao programa:

```
160 FOR I=1 TO 32
170 X=INT(RND(-TIME)*14+1)
171 Y=INT(RND(TIME)*14+1)
180 IF X=Y THEN GOTO 170
190 VPOKE B191+I,X*16+Y
200 NEXT I
```



Dessa forma você estará sorteando um número de 1 a 15 para a cor de frente a fundo (a cor transparente foi eliminada) e verificando se elas são diferentes.

Rode o programa várias vezes para ver seu efeito e observe como fica uma listagem na SCREEN 1, seu programa fica parecendo um arlequim bêbado!

## BIBLIOGRAFIA RECOMENDADA,

Coleção de Programas para MSX v.2 - páginas 18 e 19.

2.8 - USANDO 40 OU 64 COLUNAS NA SCREEN 2

Normalmente a impressão de caracteres na SCREEN 2 é semelhante à da SCREEN 1, em 32 colunas. Entretanto, com um pequeno programa em linguagem de Máquina pode-se fazer com que a impressão assemelhe-se à da SCREEN 8, em 48 colunas. Com ligeiras alterações nessa mesma rotina podemos fazê-la imprimir em 64 colunas exatamente isso que faz o programa listado a seguir. Digite-o e antes de mais nada grave-o em disco ou em fita. Depois rode-o.

```

100 SCREEN 0 : WIDTH 40
110 CLEAR 200,&HE000
120 FOR F=&HE000 TO &HE0D1
130 READ A$: POKE F,VAL("&H"+A$)
140 NEXT F
150 PRINT:
160 PRINT SPC(10);"[ 1 ] 40 COLUMNAS"
170 PRINT SPC(10);"[ 2 ] 64 COLUMNAS"
180 A$=INKEY$
190 IF A$="1" THEN 230
200 IF A$("&2") THEN 180
210 POKE &HE0AD,4
220 POKE &HE0CA,255
230 SCREEN 0 : NEW
240 DATA FE,03,C0,3A,AF,FC,FE,02
250 DATA C0,E0,46,27,5F,23,56,04
260 DATA 05,C8,1A,CD,19,E0,13,1B
270 DATA F7,F5,C5,D5,E5,FD,E5,ED
280 DATA 48,B7,FC,E0,5B,B9,FC,CD
290 DATA 39,E0,ED,43,B7,FC,ED,53
300 DATA B9,FC,FD,E1,E1,D1,C1,F1
310 DATA C9,CD,AB,00,D0,20,07,FE
320 DATA 00,28,73,FE,20,D8,6F,26
330 DATA 00,29,29,29,C5,D5,ED,5B
340 DATA 20,F9,19,11,40,FC,06,0B
350 DATA C5,D5,3A,1F,F9,C0,0L,00
360 DATA FB,D1,C1,12,13,23,10,F0
370 DATA D1,C1,3A,E9,F3,32,F2,F3
380 DATA FD,21,40,FC,D5,26,09,CB
390 DATA 7A,20,2A,CD,8F,E0,38,2B
400 DATA C5,2E,06,FD,7E,00,CB,7B
410 DATA 20,15,CD,C8,E0,38,15,CB
420 DATA 7F,2B,0C,FS,D5,E5,CD,11
430 DATA 01,CD,20,01,E1,D1,F1,07
440 DATA 03,2D,20,E2,C1,FD,23,13
450 DATA 25,20,CC,D1,21,06,00,09
460 DATA 44,4D,CD,CB,E0,D0,01,00

```

1478  
 683  
 D48  
 1476  
 1692  
 1193  
 1333  
 1334  
 1335  
 1336  
 1337  
 1338  
 1339  
 1340  
 1341  
 1342  
 1343  
 1344  
 1345  
 1346  
 1347  
 1348  
 1349  
 1350  
 1351  
 1352  
 1353  
 1354  
 1355  
 1356  
 1357  
 1358  
 1359  
 1360  
 1361  
 1362  
 1363  
 1364  
 1365  
 1366  
 1367  
 1368  
 1369  
 1370  
 1371  
 1372  
 1373  
 1374  
 1375  
 1376  
 1377  
 1378  
 1379  
 1380  
 1381  
 1382  
 1383  
 1384  
 1385  
 1386  
 1387  
 1388  
 1389  
 1390  
 1391  
 1392  
 1393  
 1394  
 1395  
 1396  
 1397  
 1398  
 1399  
 1400  
 1401  
 1402  
 1403  
 1404  
 1405  
 1406  
 1407  
 1408  
 1409  
 1410  
 1411  
 1412  
 1413  
 1414  
 1415  
 1416  
 1417  
 1418  
 1419  
 1420  
 1421  
 1422  
 1423  
 1424  
 1425  
 1426  
 1427  
 1428  
 1429  
 1430  
 1431  
 1432  
 1433  
 1434  
 1435  
 1436  
 1437  
 1438  
 1439  
 1440  
 1441  
 1442  
 1443  
 1444  
 1445  
 1446  
 1447  
 1448  
 1449  
 1450  
 1451  
 1452  
 1453  
 1454  
 1455  
 1456  
 1457  
 1458  
 1459  
 1460  
 1461  
 1462  
 1463  
 1464  
 1465  
 1466  
 1467  
 1468  
 1469  
 1470  
 1471  
 1472  
 1473  
 1474  
 1475  
 1476  
 1477  
 1478  
 1479  
 1480  
 1481  
 1482  
 1483  
 1484  
 1485  
 1486  
 1487  
 1488  
 1489  
 1490  
 1491  
 1492  
 1493  
 1494  
 1495  
 1496  
 1497  
 1498  
 1499  
 1500  
 1501  
 1502  
 1503  
 1504  
 1505  
 1506  
 1507  
 1508  
 1509  
 1510  
 1511  
 1512  
 1513  
 1514  
 1515  
 1516  
 1517  
 1518  
 1519  
 1520  
 1521  
 1522  
 1523  
 1524  
 1525  
 1526  
 1527  
 1528  
 1529  
 1530  
 1531  
 1532  
 1533  
 1534  
 1535  
 1536  
 1537  
 1538  
 1539  
 1540  
 1541  
 1542  
 1543  
 1544  
 1545  
 1546  
 1547  
 1548  
 1549  
 1550  
 1551  
 1552  
 1553  
 1554  
 1555  
 1556  
 1557  
 1558  
 1559  
 1560  
 1561  
 1562  
 1563  
 1564  
 1565  
 1566  
 1567  
 1568  
 1569  
 1570  
 1571  
 1572  
 1573  
 1574  
 1575  
 1576  
 1577  
 1578  
 1579  
 1580  
 1581  
 1582  
 1583  
 1584  
 1585  
 1586  
 1587  
 1588  
 1589  
 1590  
 1591  
 1592  
 1593  
 1594  
 1595  
 1596  
 1597  
 1598  
 1599  
 1600  
 1601  
 1602  
 1603  
 1604  
 1605  
 1606  
 1607  
 1608  
 1609  
 1610  
 1611  
 1612  
 1613  
 1614  
 1615  
 1616  
 1617  
 1618  
 1619  
 1620  
 1621  
 1622  
 1623  
 1624  
 1625  
 1626  
 1627  
 1628  
 1629  
 1630  
 1631  
 1632  
 1633  
 1634  
 1635  
 1636  
 1637  
 1638  
 1639  
 1640  
 1641  
 1642  
 1643  
 1644  
 1645  
 1646  
 1647  
 1648  
 1649  
 1650  
 1651  
 1652  
 1653  
 1654  
 1655  
 1656  
 1657  
 1658  
 1659  
 1660  
 1661  
 1662  
 1663  
 1664  
 1665  
 1666  
 1667  
 1668  
 1669  
 1670  
 1671  
 1672  
 1673  
 1674  
 1675  
 1676  
 1677  
 1678  
 1679  
 1680  
 1681  
 1682  
 1683  
 1684  
 1685  
 1686  
 1687  
 1688  
 1689  
 1690  
 1691  
 1692  
 1693  
 1694  
 1695  
 1696  
 1697  
 1698  
 1699  
 1700  
 1701  
 1702  
 1703  
 1704  
 1705  
 1706  
 1707  
 1708  
 1709  
 1710  
 1711  
 1712  
 1713  
 1714  
 1715  
 1716  
 1717  
 1718  
 1719  
 1720  
 1721  
 1722  
 1723  
 1724  
 1725  
 1726  
 1727  
 1728  
 1729  
 1730  
 1731  
 1732  
 1733  
 1734  
 1735  
 1736  
 1737  
 1738  
 1739  
 1740  
 1741  
 1742  
 1743  
 1744  
 1745  
 1746  
 1747  
 1748  
 1749  
 1750  
 1751  
 1752  
 1753  
 1754  
 1755  
 1756  
 1757  
 1758  
 1759  
 1760  
 1761  
 1762  
 1763  
 1764  
 1765  
 1766  
 1767  
 1768  
 1769  
 1770  
 1771  
 1772  
 1773  
 1774  
 1775  
 1776  
 1777  
 1778  
 1779  
 1780  
 1781



## 2.8 - CARIMBANDO SPRITES 8 x 8 NA SCREEN 2

O recurso dos SPRITES é muito útil na apresentação visual dos programas, mas eles apresentam algumas limitações, como por exemplo o fato de 5 sprites não serem impressos pelo VDP se estiverem na mesma linha.

A rotina apresentada a partir da linha 120 na listagem abaixo imprime o SPRITE 8 x 8 cujo padrão é definido pela variável SC% na coordenada dada pelas variáveis X e Y. O canal #1 para arquivos também deve estar livre para ser utilizado pela rotina.

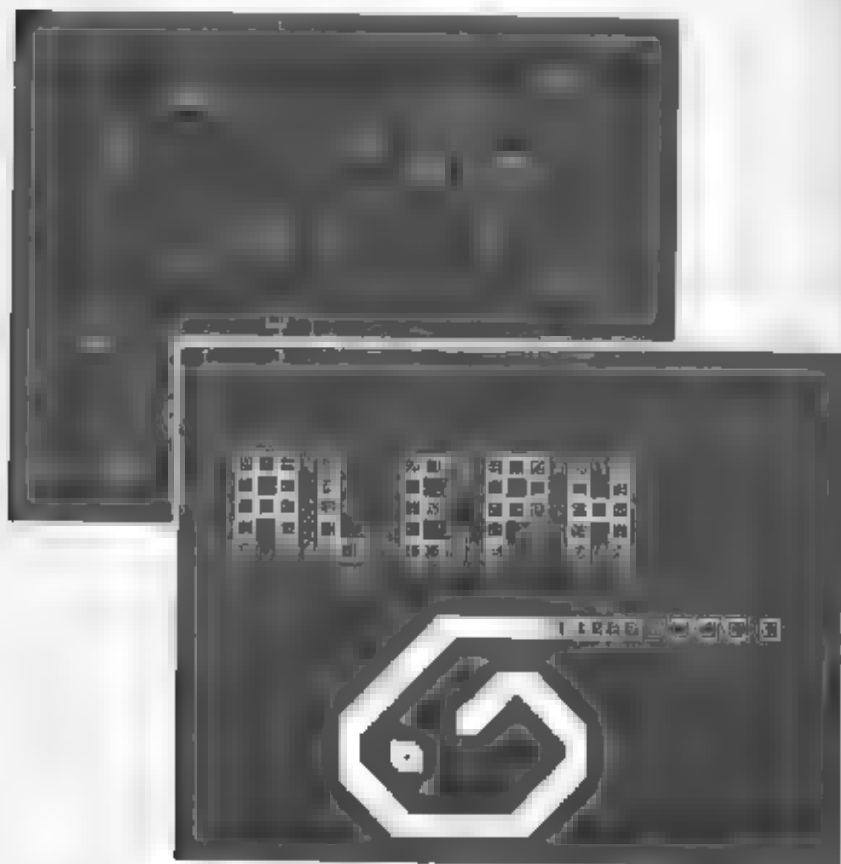
1	'*****	0000
2	'*CARIMBADOR 8x8 *	0040
3	'*BY THE DOCTOR LUZ*	0080
4	'*****	00C0
10	SL=(INP(&HA0)AND&HC0)/64	00E0
20	SCREEN 2,0:SCZ=0	0100
30	SPRITES(SCZ)=CHR\$(255)+STRING\$(6,CHR\$(255)+CHR\$(255))	0140
40	AS=INKEY\$	0180
50	PUTSPRITE 10,(X,Y),15,SCZ	01C0
60	IF AS=CHR\$(28)THEN X=X+1	0200
70	IF AS=CHR\$(29)THEN X=X-1	0240
80	IF AS=CHR\$(30)THEN Y=Y-1	0280
90	IF AS=CHR\$(31)THEN Y=Y+1	02C0
100	IF AS="F"THEN SCREEN 0:END	0300
110	IF AS=" " THEN GOSUB 130	0340
120	GOTO 40	0380
130	FOR LX=0 TO 7	03C0
140	POKE &HC200+LX,VPEEK(BASE(14)+LX+8*SCZ)	0400
150	NEXT	0440
160	POKE &HF91F,SL	0480
170	POKE &HF920,0	04C0
180	POKE &HF921,&HC0	0500
190	OPEN "GRP:" AS #1	0540
200	PRESET (X,Y+1):PRINTh1,"a"	0580
210	POKE &HF91F,&H0	05C0
220	POKE &HF920,&HBF	0600
230	POKE &HF921,&H1B	0640
240	CLOSE #1	0680
250	RETURN	06C0

TOTAL = 0000



A ideia desse carimbador de SPRITES é muito simples. A variável SC% ajuda a encontrar na VRAM o início da tabela de formação do SPRITE que é transferido para a RAM a partir do endereço &HC200.

Em seguida muda-se o conteúdo da variável do sistema CRPNT (em &HF91F) para que ao mandarmos imprimir o carácter "C" na tela gráfica não seja impressa a matriz de pontos da ROM mas o padrão do SPRITE que foi transferido a partir do endereço &HC200.



#### BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX - capítulo 4  
Programação Avançada em MSX - capítulo 2

## 2 T - CARIMBADOR DE SPRITES 16 x 16 NA SCREEN 2

O princípio de funcionamento do programa a seguir é o mesmo que o do carimbador de SPRITES 8 x 8 apresentado na dica anterior. Contudo, agora a transferência da VRAM para RAM é um pouco mais demorada, pois os SPRITES 16 x 16 são definidos por 32 bytes cada um.

```

10 *****
20 *CARIMBADOR 16x16 *
30 *BY THE DOCTOR LUZ*
40 *****
50 SL=(INP(&HAB)AND&HC0)/64
60 SCREEN 2,2:SCX=1
70 SPRITE$(SCX)=CHR$(255)+STRING$(14,CHR$(
$(8B10000000))+CHR$(255)+CHR$(255)+STRIN
G$(14,CHR$(8000000001))+CHR$(255)
80 AS=INKEY$
90 PUTSPRITE 10,(X,Y),15,SCX
100 IF AS=CHR$(28)THEN X=X+1
110 IF AS=CHR$(29)THEN X=X-1
120 IF AS=CHR$(30)THEN Y=Y-1
130 IF AS=CHR$(31)THEN Y=Y+1
140 IF AS="F"THEN SCREEN 0:END
150 IF AS=" " THEN GOSUB 170
160 GOTO 80
170 FOR LX =0 TO 31
180 POKE &HC200+LX,VPEEK(BASE(14)+LX+32*
SCX)
190 NEXT
200 POKE &HF91F,SL
210 POKE &HF920,0
220 POKE &HF921,&HC0
230 OPEN "GRP1" AS #1
240 PRESET (X,Y+1):PRINT#1,"0B"
250 PRESET (X,Y+9):PRINT#1,"AC"
260 POKE &HF91F,&H0
270 POKE &HF920,&HBF
280 POKE &HF921,&H1B
290 CLOSE #1
300 RETURN

```

\*\*\*\*\* = 88F

### BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX - capítulo 4  
 Programação Avançada em MSX - capítulo 2.

## 2 U - ARMAZENANDO TELAS NA RAM

O programa em BASIC listado a seguir gera na memória do micro a partir do endereço &HE000 uma rotina em Linguagem de Máquina que permite o armazenamento da tela na RAM e sua posterior recuperação de volta para a VRAM.

A primeira parte do programa é a responsável pela geração da rotina de transferência. A segunda parte é um exemplo de como a rotina em LM deve ser usada.

Observe a incrível velocidade com que a rotina de transferência consegue armazenar (e recuperar) os 16 kbytes da VRAM (o fm BASIC essa mesma transferência se fosse possível certamente demoraria algo em torno de 100 vezes mais).

100 FOR F=&HE000 TO &HE03E	100
110 READ A% : POKE F,VAL("&H"+A%)	101
120 NEXT F	102
130 DEFUSR0=&HE000 : DEFUSR1=&HE011	103
140 DATA 10,20,00,00,00,11,00	104
150 DATA 40,01,FF,3F,CD,59,00,10	105
160 DATA 24,FD,02,00,21,00,40,11	106
170 DATA 00,00,01,FF,3F,CD,5C,00	107
180 DATA 10,10,FB,0B,A0,47,CB,3F	108
190 DATA CB,3F,CB,3F,CB,3F,80,E6	109
200 DATA FC,D3,AB,FB,C9,43,DB,AB	110
210 DATA E6,FD,DJ,AB,FB,C9,A2,00	111
220 REM -- F exemplo de utilização -	112
230 COLOR 1,7,5 : SCREEN 2	113
240 FOR F=0 TO 1 STEP 10	114
250 CIRC F(128,80),80,1,,,80/F	115
260 CIRCLE(128,80),80,1,,,F/80	116
270 NEXT F	117
280 LINE (128,160)-(128,0)	118
290 LINE (48,80)-(200,0)	119
300 POKE 0,USR0(0) : ' VRAM para RAM	120
310 SCREEN 0 : WIDTH 30 : LOCATE 10,10	121
320 PRINT "Digite RETURN":A%=INPUT\$(1)	122
330 SCREEN 2	123
340 POKE 0,USR1(0) : ' RAM para VRAM	124
350 GOTO 350	125

## BIBLIOGRAFIA RECOMENDADA,

Aprofundando-se no MSX - capítulos 0, 3 e 4  
Programação Avançada em MSX - capítulo 2  
Criação de Programas para MSX v.1 - páginas 31 a 54.

## 2 V - USANDO A VRAM PARA DADOS

O vídeo do MSX é controlado por um circuito especial chamado VDP que tem a sua disposição 16 Kbytes de memória chamada Video RAM (VRAM).

O VDP pode usar a VRAM de quatro formas diferentes correspondentes às SCREEN 0, 1, 2 e 3 do MSX.

Para cada SCREEN a VRAM é dividida de forma diferente e sempre sobram algumas áreas não ocupadas. A seguir apresentamos uma tabela onde relacionamos a área não usada da VRAM para cada SCREEN do MSX.

SCREEN	ÁREA NÃO USADA
0	960 até 2047 4096 até 16383
1	2048 até 8143 7840 até 8191 8224 até 14335
2	7840 até 8191
3	3584 até 8812 7840 até 14335

Todas as áreas livres da VRAM podem ser usadas para armazenamento de dados. Isso pode ser feito tanto com rotinas em linguagem de Máquina como com o comando VPOKE do BASIC.

Vamos ilustrar de forma bem simples como se pode usar a VRAM para dados. O que é e execute o programa a seguir.

```
10 SCREEN 0 : WIDTH 38
20 AS "DADOS EXEMPLOS PARA A VRAM"
30 VPOKE 4096, LEN(AS)
40 FOR F=1 TO LEN(AS)
50   VPOKE 4096+F, ASC(MID$(AS, F, 1))
60 NEXT F
```

207  
C-3  
1100  
1100  
207  
207  
207

Com isso os dados da string AS estarão na VRAM. Agora você pode comandar NEW apagando o programa e as

variáveis da memória RAM normal do micro

Para recuperar os dados que foram passados para a VRAM digite e rode o programa abaixo

```
10 SCREEN 0 : WIDTH 39
20 X=VPEEK(4096)
30 A$=""
40 FOR F=1 TO X
50   A$=A$+CHR$(VPEEK(4096+F))
60 NEXT F
70 PRINT A$
```

298  
629  
858  
C88  
1412  
1468  
1868

Com isso A\$ estará novamente com os dados que havíamos gravado na VRAM.



#### BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX páginas 105 e 106

## 2.M - PESQUISADOR DE DESENHOS

O programa apresentado adiante é um editor de caracteres para telas gráficas. Digite-o, confira-o e grave-o. Depois, rode o programa.

Ao ser executado ele permite que o usuário "vasculhe" a memória do micro a procura de um desenho para te a gráfica. Os dois terços superiores da tela são usados para mostrar os desenhos dos caracteres da região da memória pesquisada tal qual eles apareceriam na figura da tela gráfica. O terço inferior da tela é usado para mostrar um único caractere em destaque. Esse caractere pode ser redesenhado e corresponde à ampliação das posições indicadas por um pequeno quadrado no centro da tela.

Assim que o programa é carregado as teclas de setas permitem mover a região da memória pesquisada lembrando-se que de &H0000 a &H7FFF existe ROM e a partir de &H8000 existe RAM. As funções das teclas de setas estão resumidas abaixo.

- pesquisa um byte para a frente
- SHIFT + ► pesquisa 8 bytes para frente (1 caractere),
- ◄ pesquisa um byte para trás
- SHIFT + ◄ pesquisa 8 bytes para trás (1 caractere),
- ▲ pesquisa 256 bytes para frente (1 linha),
- SHIFT + ▲ pesquisa 2048 bytes para frente (1/3 de tela),
- ▼ pesquisa 256 bytes para trás (1 linha)
- SHIFT + ▼ pesquisa 2048 bytes para trás (1/3 de tela).

Uma vez escolhida a região de memória a ser pesquisada, pode-se editar o caractere em destaque. Para isso deve-se digitar a tecla SELECT. Ela permite a passagem do modo "seleção" para o modo "edição". Pressionando SELECT a primeira vez um pequeno cursor surgirá no caractere em destaque na parte inferior da tela. Pode-se movê-lo com as teclas de setas. Para alterar um ponto, basta pressionar a barra de espaços; se ele estiver apagado, será aceso e, se estiver aceso, será apagado.

Com o caractere já editado, pode-se retornar do modo de edição digitando mais uma vez a tecla SELECT.

Para sair do programa e retornar ao BASIC basta estar fora do modo de edição e pressionar a barra de espaços

Se ao retornar ao BASIC a tela ficar toda com uma só cor, use um comando:

color 15,1,1

Isso deverá resolver o problema

```

100 C=VAR 200 &H0000
105 FOR F% &H0000 TO &H032A
110   READ A% : POKE F%,VAL("&H"+A%)
115 NEXT F%
120 B=AVE ACHADE5,"BIN",&H0000,&H032A
125 DEFUN R0 &H0000 : END
130 DATA 0E,0F,1E,EB,F3,CD,72,00
135 DATA 1E,01,1E,B0,FB,21,FA,D2
140 DATA 1D,5E,C2,F3,01,10,00,1D
145 DATA 1E,00,21,0A,D3,FD,1E,1D
150 DATA 13,01,03,00,CD,5C,00,1D
155 DATA 03,D3,E5,CD,07,D2,E1,2A
160 DATA 1D,D1,11,00,00,01,00,10
165 DATA CD,5C,00,06,01,21,E0,FB
170 DATA 7E,C8,4E,20,02,06,00,21
175 DATA 23,7E,C0,7E,20,0C,2A,FB
180 DATA 12,2E,10,1D,22,FB,D2,C3
185 DATA 22,D0,C0,67,20,0C,2A,FB
190 DATA D2,2B,10,FD,22,FB,D2,C3
195 DATA 22,D0,C0,77,20,0E,2A,FB
200 DATA D2,11,00,01,19,10,FD,22
205 DATA FB,D1,C3,27,D0,C0,AE,20
210 DATA 11,2A,FB,D2,AE,11,00,01
215 DATA 1D,5E,10,FB,22,FB,D2,C3
220 DATA 22,D0,21,E2,FB,7E,C0,47
225 DATA 20,01,12,2B,7E,C0,77,C2
230 DATA 70,D0,1D,C0,00,21,11,D3
235 DATA 76,04,20,20,JA,E2,FB,07
240 DATA 30,01,1E,D0,02,07,2B,30
245 DATA 03,CD,0B,D2,07,70,03,CD
250 DATA E1,D2,07,23,30,01,FD,E1
255 DATA D2,7E,FE,60,30,02,76,60
260 DATA 1E,B3,38,02,36,81,2D,7E
265 DATA 1E,BB,30,02,76,80,FE,AE
270 DATA 11,02,36,AE,01,0B,00,ED
275 DATA 50,C0,F3,21,0A,D3,CD,5C
280 DATA 00,C0,E7,D2,21,F2,FB,7E
285 DATA 0E,DA,7E,D1,2A,FB,D2,01
290 DATA 80,0B,09,3A,0E,D3,FE,90

```

0000  
 0001  
 0002  
 0003  
 0004  
 0005  
 0006  
 0007  
 0008  
 0009  
 0010  
 0011  
 0012  
 0013  
 0014  
 0015  
 0016  
 0017  
 0018  
 0019  
 0020  
 0021  
 0022  
 0023  
 0024  
 0025  
 0026  
 0027  
 0028  
 0029  
 0030  
 0031  
 0032  
 0033  
 0034  
 0035  
 0036  
 0037  
 0038  
 0039  
 0040  
 0041  
 0042  
 0043  
 0044  
 0045  
 0046  
 0047  
 0048  
 0049  
 0050  
 0051  
 0052  
 0053  
 0054  
 0055  
 0056  
 0057  
 0058  
 0059  
 0060  
 0061  
 0062  
 0063  
 0064  
 0065  
 0066  
 0067  
 0068  
 0069  
 0070  
 0071  
 0072  
 0073  
 0074  
 0075  
 0076  
 0077  
 0078  
 0079  
 0080  
 0081  
 0082  
 0083  
 0084  
 0085  
 0086  
 0087  
 0088  
 0089  
 0090  
 0091  
 0092  
 0093  
 0094  
 0095  
 0096  
 0097  
 0098  
 0099  
 0100

295 DATA 38,01,23,FE,95,38,01,23  
 300 DATA FE,9A,38,01,23,FE,9F,38  
 305 DATA 01,23,FE,A4,38,01,23,FE  
 310 DATA A7,38,01,23,FE,AE,38,01  
 315 DATA 23,FE,B3,38,01,23,04,00  
 320 DATA 3A,0F,D3,FE,60,20,04,CB  
 325 DATA FB,18,36,FE,65,20,04,CB  
 330 DATA F0,18,2E,FE,6A,20,04,CB  
 335 DATA E8,18,26,FE,6F,20,04,CB  
 340 DATA E0,18,1E,FE,74,20,04,CB  
 345 DATA D0,18,16,FE,79,20,04,CB  
 350 DATA D8,18,0E,FE,7F,20,04,CB  
 355 DATA C8,18,04,FE,83,20,04,CB  
 360 DATA C0,7E,A0,77,2A,FB,D2,01  
 365 DATA 80,00,09,CD,07,D1,21,E1  
 370 DATA FB,7E,CB,77,C2,A1,D0,00  
 375 DATA 21,0E,D3,DD,36,00,88,DD  
 380 DATA 21,DD,36,00,60,DD,23,DD  
 385 DATA 21,DD,36,00,00,CD,00,00  
 390 DATA 01,00,00,FD,5B,CD,13,21  
 395 DATA 0A,D3,CD,5F,00,CD,E7,D3  
 400 DATA C3,3B,D0,21,12,D3,CD,0E  
 405 DATA 4B,3E,0F,32,E9,F3,3E,20  
 410 DATA 31,07,FF,3E,BA,32,07,EE  
 415 DATA 2A,1B,D2,01,80,0B,09,E5  
 420 DATA 71,CD,CB,02,CD,CD,01,E1  
 425 DATA E5,7C,CD,CD,D2,E1,15,7D  
 430 DATA CD,CB,D2,1D,CD,D2,E1,E5  
 435 DATA 7D,CD,CD,D2,3F,01,CD,8D  
 440 DATA 00,3E,57,CD,8D,00,3E,01  
 445 DATA CD,8D,00,3F,57,CD,8D,00  
 450 DATA 3E,CF,CD,8D,00,E1,C9,01  
 455 DATA 00,00,09,E5,AF,ED,42,C1  
 460 DATA 7F,21,E9,F3,36,01,07,30  
 465 DATA 02,36,0F,21,87,FC,36,60  
 470 DATA 21,84,00,41,28,2B,28,2B  
 475 DATA 2B,10,F9,27,89,FC,08,3E  
 480 DATA D3,CD,8D,00,08,21,E9,F3  
 485 DATA 76,01,07,30,02,36,01,21  
 490 DATA B7,FC,36,65,0B,CD,DD,00  
 495 DATA 0B,21,E9,F3,36,01,07,30  
 500 DATA 02,36,0F,21,B7,FC,36,6A  
 505 DATA 0B,CD,8D,00,08,21,E9,F3  
 510 DATA 76,01,07,30,02,36,01,21  
 515 DATA B7,FC,36,6F,0B,CD,8D,00  
 520 DATA 0B,21,E9,F3,36,01,07,30  
 525 DATA 02,36,0F,21,B7,FC,36,74  
 530 DATA 0B,CD,8D,00,08,21,E9,F3  
 535 DATA 36,01,07,30,02,36,01,21

12000  
 12001  
 12002  
 12003  
 12004  
 12005  
 12006  
 12007  
 12008  
 12009  
 12010  
 12011  
 12012  
 12013  
 12014  
 12015  
 12016  
 12017  
 12018  
 12019  
 12020  
 12021  
 12022  
 12023  
 12024  
 12025  
 12026  
 12027  
 12028  
 12029  
 12030  
 12031  
 12032  
 12033  
 12034  
 12035  
 12036  
 12037  
 12038  
 12039  
 12040  
 12041  
 12042  
 12043  
 12044  
 12045  
 12046  
 12047  
 12048  
 12049  
 12050  
 12051  
 12052  
 12053  
 12054  
 12055  
 12056  
 12057  
 12058  
 12059  
 12060  
 12061  
 12062  
 12063  
 12064  
 12065  
 12066  
 12067  
 12068  
 12069  
 12070  
 12071  
 12072  
 12073  
 12074  
 12075  
 12076  
 12077  
 12078  
 12079  
 12080  
 12081  
 12082  
 12083  
 12084  
 12085  
 12086  
 12087  
 12088  
 12089  
 12090  
 12091  
 12092  
 12093  
 12094  
 12095  
 12096  
 12097  
 12098  
 12099  
 12100  
 12101  
 12102  
 12103  
 12104  
 12105  
 12106  
 12107  
 12108  
 12109  
 12110  
 12111  
 12112  
 12113  
 12114  
 12115  
 12116  
 12117  
 12118  
 12119  
 12120  
 12121  
 12122  
 12123  
 12124  
 12125  
 12126  
 12127  
 12128  
 12129  
 12130  
 12131  
 12132  
 12133  
 12134  
 12135  
 12136  
 12137  
 12138  
 12139  
 12140  
 12141  
 12142  
 12143  
 12144  
 12145  
 12146  
 12147  
 12148  
 12149  
 12150  
 12151  
 12152  
 12153  
 12154  
 12155  
 12156  
 12157  
 12158  
 12159  
 12160  
 12161  
 12162  
 12163  
 12164  
 12165  
 12166  
 12167  
 12168  
 12169  
 12170  
 12171  
 12172  
 12173  
 12174  
 12175  
 12176  
 12177  
 12178  
 12179  
 12180  
 12181  
 12182  
 12183  
 12184  
 12185  
 12186  
 12187  
 12188  
 12189  
 12190  
 12191  
 12192  
 12193  
 12194  
 12195  
 12196  
 12197  
 12198  
 12199  
 12200  
 12201  
 12202  
 12203  
 12204  
 12205  
 12206  
 12207  
 12208  
 12209  
 12210  
 12211  
 12212  
 12213  
 12214  
 12215  
 12216  
 12217  
 12218  
 12219  
 12220  
 12221  
 12222  
 12223  
 12224  
 12225  
 12226  
 12227  
 12228  
 12229  
 12230  
 12231  
 12232  
 12233  
 12234  
 12235  
 12236  
 12237  
 12238  
 12239  
 12240  
 12241  
 12242  
 12243  
 12244  
 12245  
 12246  
 12247  
 12248  
 12249  
 12250  
 12251  
 12252  
 12253  
 12254  
 12255  
 12256  
 12257  
 12258  
 12259  
 12260  
 12261  
 12262  
 12263  
 12264  
 12265  
 12266  
 12267  
 12268  
 12269  
 12270  
 12271  
 12272  
 12273  
 12274  
 12275  
 12276  
 12277  
 12278  
 12279  
 12280  
 12281  
 12282  
 12283  
 12284  
 12285  
 12286  
 12287  
 12288  
 12289  
 12290  
 12291  
 12292  
 12293  
 12294  
 12295  
 12296  
 12297  
 12298  
 12299  
 12300  
 12301  
 12302  
 12303  
 12304  
 12305  
 12306  
 12307  
 12308  
 12309  
 12310  
 12311  
 12312  
 12313  
 12314  
 12315  
 12316  
 12317  
 12318  
 12319  
 12320  
 12321  
 12322  
 12323  
 12324  
 12325  
 12326  
 12327  
 12328  
 12329  
 12330  
 12331  
 12332  
 12333  
 12334  
 12335  
 12336  
 12337  
 12338  
 12339  
 12340  
 12341  
 12342  
 12343  
 12344  
 12345  
 12346  
 12347  
 12348  
 12349  
 12350  
 12351  
 12352  
 12353  
 12354  
 12355  
 12356  
 12357  
 12358  
 12359  
 12360  
 12361  
 12362  
 12363  
 12364  
 12365  
 12366  
 12367  
 12368  
 12369  
 12370  
 12371  
 12372  
 12373  
 12374  
 12375  
 12376  
 12377  
 12378  
 12379  
 12380  
 12381  
 12382  
 12383  
 12384  
 12385  
 12386  
 12387  
 12388  
 12389  
 12390  
 12391  
 12392  
 12393  
 12394  
 12395  
 12396  
 12397  
 12398  
 12399  
 12400  
 12401  
 12402  
 12403  
 12404  
 12405  
 12406  
 12407  
 12408  
 12409  
 12410  
 12411  
 12412  
 12413  
 12414  
 12415  
 12416  
 12417  
 12418  
 12419  
 12420  
 12421  
 12422  
 12423  
 12424  
 12425  
 12426  
 12427  
 12428  
 12429  
 12430  
 12431  
 12432  
 12433  
 12434  
 12435  
 12436  
 12437  
 12438  
 12439  
 12440  
 12441  
 12442  
 12443  
 12444  
 12445  
 12446  
 12447  
 12448  
 12449  
 12450  
 12451  
 12452  
 12453  
 12454  
 12455  
 12456  
 12457  
 12458  
 12459  
 12460  
 12461  
 12462  
 12463  
 12464  
 12465  
 12466  
 12467  
 12468  
 12469  
 12470  
 12471  
 12472  
 12473  
 12474  
 12475  
 12476  
 12477  
 12478  
 12479  
 12480  
 12481  
 12482  
 12483  
 12484  
 12485  
 12486  
 12487  
 12488  
 12489  
 12490  
 12491  
 12492  
 12493  
 12494  
 12495  
 12496  
 12497  
 12498  
 12499  
 12500  
 12501  
 12502  
 12503  
 12504  
 12505  
 12506  
 12507  
 12508  
 12509  
 12510  
 12511  
 12512  
 12513  
 12514  
 12515  
 12516  
 12517  
 12518  
 12519  
 12520  
 12521  
 12522  
 12523  
 12524  
 12525  
 12526  
 12527  
 12528  
 12529  
 12530  
 12531  
 12532  
 12533  
 12534  
 12535  
 12536  
 12537  
 12538  
 12539  
 12540  
 12541  
 12542  
 12543  
 12544  
 12545  
 12546  
 12547  
 12548  
 12549  
 12550  
 12551  
 12552  
 12553  
 12554  
 12555  
 12556  
 12557  
 12558  
 12559  
 12560  
 12561  
 12562  
 12563  
 12564  
 12565  
 12566  
 12567  
 12568  
 12569  
 12570  
 12571  
 12572  
 12573  
 12574  
 12575  
 12576  
 12577  
 12578  
 12579  
 12580  
 12581  
 12582  
 12583  
 12584  
 12585  
 12586  
 12587  
 12588  
 12589  
 12590  
 12591  
 12592  
 12593  
 12594  
 12595  
 12596  
 12597  
 12598  
 12599  
 12600  
 12601  
 12602  
 12603  
 12604  
 12605  
 12606  
 12607  
 12608  
 12609  
 12610  
 12611  
 12612  
 12613  
 12614  
 12615  
 12616  
 12617  
 12618  
 12619  
 12620  
 12621  
 12622  
 12623  
 12624  
 12625  
 12626  
 12627  
 12628  
 12629  
 12630  
 12631  
 12632  
 12633  
 12634  
 12635  
 12636  
 12637  
 12638  
 12639  
 12640  
 12641  
 12642  
 12643  
 12644  
 12645  
 12646  
 12647  
 12648  
 12649  
 12650  
 12651  
 12652  
 12653  
 12654  
 12655  
 12656  
 12657  
 12658  
 12659  
 12660  
 12661  
 12662  
 12663  
 12664  
 12665  
 12666  
 12667  
 12668  
 12669  
 12670  
 12671  
 12672  
 12673  
 12674  
 12675  
 12676  
 12677  
 12678  
 12679  
 12680  
 12681  
 12682  
 12683  
 12684  
 12685  
 12686  
 12687  
 12688  
 12689  
 12690  
 12691  
 12692  
 12693  
 12694  
 12695  
 12696  
 12697  
 12698  
 12699  
 12700  
 12701  
 12702  
 12703  
 12704  
 12705  
 12706  
 12707  
 12708  
 12709  
 12710  
 12711  
 12712  
 12713  
 12714  
 12715  
 12716  
 12717  
 12718  
 12719  
 12720  
 12721  
 12722  
 12723  
 12724  
 12725  
 12726  
 12727  
 12728  
 12729  
 12730  
 12731  
 12732  
 12733  
 12734  
 12735  
 12736  
 12737  
 12738  
 12739  
 12740  
 12741  
 12742  
 12743  
 12744  
 12745  
 12746  
 12747  
 12748  
 12749  
 12750  
 12751  
 12752  
 12753  
 12754  
 12755  
 12756  
 12757  
 12758  
 12759  
 12760  
 12761  
 12762  
 12763  
 12764  
 12765  
 12766  
 12767  
 12768  
 12769  
 12770  
 12771  
 12772  
 12773  
 12774  
 12775  
 12776  
 12777  
 12778  
 12779  
 12780  
 12781  
 12782  
 12783  
 12784  
 12785  
 12786  
 12787  
 12788  
 12789  
 12790  
 12791  
 12792  
 12793  
 12794  
 12795  
 12796  
 12797  
 12798  
 12799  
 12800  
 12801  
 12802  
 12803  
 12804  
 12805  
 12806  
 12807  
 12808  
 12809  
 12810  
 12811  
 12812  
 12813  
 12814  
 12815  
 12816  
 12817  
 12818  
 12819  
 12820  
 12821  
 12822  
 12823  
 12824  
 12825  
 12826  
 12827  
 12828  
 12829  
 12830  
 12831  
 12832  
 12833  
 12834  
 12835  
 12836  
 12837  
 12838  
 12839  
 12840  
 12841  
 12842  
 12843  
 12844  
 12845  
 12846  
 12847  
 12848  
 12849  
 12850  
 12851  
 12852  
 12853  
 12854  
 12855  
 12856  
 12857  
 12858  
 12859  
 12860  
 12861  
 12862  
 12863  
 12864  
 12865  
 12866  
 12867  
 12868  
 12869  
 12870  
 12871  
 12872  
 12873  
 12874  
 12875  
 12876  
 12877  
 12878  
 12879  
 12880  
 12881  
 12882  
 12883  
 12884  
 12885  
 12886  
 12887  
 12888  
 12889  
 12890  
 12891  
 12892  
 12893  
 12894  
 12895  
 12896  
 12897  
 12898  
 12899  
 12900  
 12901  
 12902  
 12903  
 12904  
 12905  
 12906  
 12907  
 12908  
 12909  
 12910  
 12911  
 12912  
 12913  
 12914  
 12915  
 12916  
 12917  
 12918  
 12919  
 12920  
 12921  
 12922  
 12923  
 12924  
 12925  
 12926  
 12927  
 12928  
 12929  
 12930  
 12931  
 12932  
 12933  
 12934  
 12935  
 12936  
 12937  
 12938  
 12939  
 12940  
 12941  
 12942  
 12943  
 12944  
 12945  
 12946  
 12947  
 12948  
 12949  
 12950  
 12951  
 12952  
 12953  
 12954  
 12955  
 12956  
 12957  
 12958  
 12959  
 12960  
 12961  
 12962  
 12963  
 12964  
 12965  
 12966  
 12967  
 12968  
 12969  
 12970  
 12971  
 12972  
 12973  
 12974  
 12975  
 12976  
 12977  
 12978  
 12979  
 12980  
 12981  
 12982  
 12983  
 12984  
 12985  
 12986  
 12987  
 12988  
 12989  
 12990  
 12991  
 12992  
 12993  
 12994  
 12995  
 12996  
 12997  
 12998  
 12999  
 13000



```

540 DATA B7,FC,36,79,08,CD,8D,00
545 DATA 08,21,E9,F3,36,01,07,30
550 DATA 02,36,0F,21,B7,FC,36,7E
555 DATA 08,CD,8D,00,08,21,E9,F3
560 DATA 36,01,07,30,02,36,0F,21
565 DATA B7,FC,36,83,08,CD,8D,00
570 DATA 08,C1,E1,0D,C2,08,D2,C9
575 DATA 0F,0F,0F,0F,C9,E6,0F,C4
580 DATA 30,FE,3A,38,02,C6,07,CD
585 DATA 8D,00,C9,35,35,35,35,35
590 DATA C9,34,34,34,34,34,C9,D9
595 DATA 21,29,D3,46,C5,06,00,C5
600 DATA C1,10,FC,C1,10,F6,D9,C9
605 DATA 00,B0,FF,FF,C3,C3,C3,C3
610 DATA FF,FF,F8,88,88,88,F8,00
615 DATA 00,00,3F,80,00,06,88,60
620 DATA 01,00,28,34,30,2C,31,33
625 DATA 30,29,F2,28,37,32,2C,31
630 DATA 34,36,29,2C,31,2C,42,46
635 DATA 00,32,55

```

```

8991
8993
9004
9005
9006
9007
9008
9009
9010
9011
9012
9013
9014
9015
9016
9017
9018
9019
9020
9021
9022
9023
9024
9025
9026
9027
9028
9029
9030
9031
9032
9033
9034
9035
9036
9037
9038
9039
9040
9041
9042
9043
9044
9045
9046
9047
9048
9049
9050
9051
9052
9053
9054
9055
9056
9057
9058
9059
9060
9061
9062
9063
9064
9065
9066
9067
9068
9069
9070
9071
9072
9073
9074
9075
9076
9077
9078
9079
9080
9081
9082
9083
9084
9085
9086
9087
9088
9089
9090
9091
9092
9093
9094
9095
9096
9097
9098
9099
9100
9101
9102
9103
9104
9105
9106
9107
9108
9109
9110
9111
9112
9113
9114
9115
9116
9117
9118
9119
9120
9121
9122
9123
9124
9125
9126
9127
9128
9129
9130
9131
9132
9133
9134
9135
9136
9137
9138
9139
9140
9141
9142
9143
9144
9145
9146
9147
9148
9149
9150
9151
9152
9153
9154
9155
9156
9157
9158
9159
9160
9161
9162
9163
9164
9165
9166
9167
9168
9169
9170
9171
9172
9173
9174
9175
9176
9177
9178
9179
9180
9181
9182
9183
9184
9185
9186
9187
9188
9189
9190
9191
9192
9193
9194
9195
9196
9197
9198
9199
9200
9201
9202
9203
9204
9205
9206
9207
9208
9209
9210
9211
9212
9213
9214
9215
9216
9217
9218
9219
9220
9221
9222
9223
9224
9225
9226
9227
9228
9229
9230
9231
9232
9233
9234
9235
9236
9237
9238
9239
9240
9241
9242
9243
9244
9245
9246
9247
9248
9249
9250
9251
9252
9253
9254
9255
9256
9257
9258
9259
9260
9261
9262
9263
9264
9265
9266
9267
9268
9269
9270
9271
9272
9273
9274
9275
9276
9277
9278
9279
9280
9281
9282
9283
9284
9285
9286
9287
9288
9289
9290
9291
9292
9293
9294
9295
9296
9297
9298
9299
9300
9301
9302
9303
9304
9305
9306
9307
9308
9309
9310
9311
9312
9313
9314
9315
9316
9317
9318
9319
9320
9321
9322
9323
9324
9325
9326
9327
9328
9329
9330
9331
9332
9333
9334
9335
9336
9337
9338
9339
9340
9341
9342
9343
9344
9345
9346
9347
9348
9349
9350
9351
9352
9353
9354
9355
9356
9357
9358
9359
9360
9361
9362
9363
9364
9365
9366
9367
9368
9369
9370
9371
9372
9373
9374
9375
9376
9377
9378
9379
9380
9381
9382
9383
9384
9385
9386
9387
9388
9389
9390
9391
9392
9393
9394
9395
9396
9397
9398
9399
9400
9401
9402
9403
9404
9405
9406
9407
9408
9409
9410
9411
9412
9413
9414
9415
9416
9417
9418
9419
9420
9421
9422
9423
9424
9425
9426
9427
9428
9429
9430
9431
9432
9433
9434
9435
9436
9437
9438
9439
9440
9441
9442
9443
9444
9445
9446
9447
9448
9449
9450
9451
9452
9453
9454
9455
9456
9457
9458
9459
9460
9461
9462
9463
9464
9465
9466
9467
9468
9469
9470
9471
9472
9473
9474
9475
9476
9477
9478
9479
9480
9481
9482
9483
9484
9485
9486
9487
9488
9489
9490
9491
9492
9493
9494
9495
9496
9497
9498
9499
9500
9501
9502
9503
9504
9505
9506
9507
9508
9509
9510
9511
9512
9513
9514
9515
9516
9517
9518
9519
9520
9521
9522
9523
9524
9525
9526
9527
9528
9529
9530
9531
9532
9533
9534
9535
9536
9537
9538
9539
9540
9541
9542
9543
9544
9545
9546
9547
9548
9549
9550
9551
9552
9553
9554
9555
9556
9557
9558
9559
9560
9561
9562
9563
9564
9565
9566
9567
9568
9569
9570
9571
9572
9573
9574
9575
9576
9577
9578
9579
9580
9581
9582
9583
9584
9585
9586
9587
9588
9589
9590
9591
9592
9593
9594
9595
9596
9597
9598
9599
9600
9601
9602
9603
9604
9605
9606
9607
9608
9609
9610
9611
9612
9613
9614
9615
9616
9617
9618
9619
9620
9621
9622
9623
9624
9625
9626
9627
9628
9629
9630
9631
9632
9633
9634
9635
9636
9637
9638
9639
9640
9641
9642
9643
9644
9645
9646
9647
9648
9649
9650
9651
9652
9653
9654
9655
9656
9657
9658
9659
9660
9661
9662
9663
9664
9665
9666
9667
9668
9669
9670
9671
9672
9673
9674
9675
9676
9677
9678
9679
9680
9681
9682
9683
9684
9685
9686
9687
9688
9689
9690
9691
9692
9693
9694
9695
9696
9697
9698
9699
9700
9701
9702
9703
9704
9705
9706
9707
9708
9709
9710
9711
9712
9713
9714
9715
9716
9717
9718
9719
9720
9721
9722
9723
9724
9725
9726
9727
9728
9729
9730
9731
9732
9733
9734
9735
9736
9737
9738
9739
9740
9741
9742
9743
9744
9745
9746
9747
9748
9749
9750
9751
9752
9753
9754
9755
9756
9757
9758
9759
9760
9761
9762
9763
9764
9765
9766
9767
9768
9769
9770
9771
9772
9773
9774
9775
9776
9777
9778
9779
9780
9781
9782
9783
9784
9785
9786
9787
9788
9789
9790
9791
9792
9793
9794
9795
9796
9797
9798
9799
9800
9801
9802
9803
9804
9805
9806
9807
9808
9809
9810
9811
9812
9813
9814
9815
9816
9817
9818
9819
9820
9821
9822
9823
9824
9825
9826
9827
9828
9829
9830
9831
9832
9833
9834
9835
9836
9837
9838
9839
9840
9841
9842
9843
9844
9845
9846
9847
9848
9849
9850
9851
9852
9853
9854
9855
9856
9857
9858
9859
9860
9861
9862
9863
9864
9865
9866
9867
9868
9869
9870
9871
9872
9873
9874
9875
9876
9877
9878
9879
9880
9881
9882
9883
9884
9885
9886
9887
9888
9889
9890
9891
9892
9893
9894
9895
9896
9897
9898
9899
9900
9901
9902
9903
9904
9905
9906
9907
9908
9909
9910
9911
9912
9913
9914
9915
9916
9917
9918
9919
9920
9921
9922
9923
9924
9925
9926
9927
9928
9929
9930
9931
9932
9933
9934
9935
9936
9937
9938
9939
9940
9941
9942
9943
9944
9945
9946
9947
9948
9949
9950
9951
9952
9953
9954
9955
9956
9957
9958
9959
9960
9961
9962
9963
9964
9965
9966
9967
9968
9969
9970
9971
9972
9973
9974
9975
9976
9977
9978
9979
9980
9981
9982
9983
9984
9985
9986
9987
9988
9989
9990
9991
9992
9993
9994
9995
9996
9997
9998
9999

```

DATE = 8/04



## BIBLIOGRAFIA RECOMENDADA.

Aprofundando se no MSX capítulo 4.  
Programação Avançada em MSX - capítulo 2

## 2.1 - IMAGENS INSTANTÂNEAS

Muitas vezes a produção de uma tela por um programa em BASIC é demorada demais, principalmente se estamos trabalhando com a SCREEN 2.

Podemos fazer com que a imagem não seja mostrada durante a confecção do desenho, mas somente após ele estar pronto.

Uma forma bem simples de conseguir isso é usando o comando COLOR como exemplificamos a seguir.

```

100 COLOR 0,9,9 : SCREEN 2
110 REM          - tela exemplo -----
120 FOR F=0 TO 125 STEP 10
130     G=80-F*80/125
140     LINE (F,80)-(125,G)
150     LINE (125,G)-(250-F,80)
160     LINE (F,80)-(125,160-G)
170     LINE (125,160-G)-(250-F,80)
180 NEXT F
190 COLOR , ,1
200 BEEP : GOTO 200

```

Na linha 100 fazemos com que o cor da frente, de fundo e da borda da tela sejam iguais, possibilitando a visão de que está sendo feito na tela. A linha 190 só é executada após o desenho estar terminado e coloca as cores na tela de uma só vez. Dessa forma, não vemos o desenho ser feito.

Experimente executar o programa novamente  
e'll nando o comando COLOR da linha 100. Você verá o  
desenho sendo elaborado na tela.

Uma outra forma de conseguir o mesmo efeito é usando as rotinas do BIOS DISKSR (8H41) e ENASCR (18H44) como no programa a seguir.

```

10 DEFUSR0 &H41 : DEFUSR1=&H44
20 POKE 0,USR0(0)
30 FOR F=1 TO 24
40   PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
50 NEXT F
60 POKE 0,USR1(0)

```

## BIBLIOGRAFIA RECOMENDADA.

Apofundando-se no MSX - página 147.

Coleção de Programas para MSX v 2 - páginas 42 e 68

## 2 Y - IMPRESSOR EM TAMANHO DUPLA NA SCREEN 2

Aqui vai um programinha em linguagem de máquina que permite ao seu programa em BASIC imprimir qualquer mensagem em tamanho dobrado na SCREEN 2. A grande utilidade do e é a de preencher a lacuna existente entre as SCREEN's 2 e 3, já que na primeira impressão é em tamanho natural e na segunda em letras tiranossáuricas.

A utilização do programa é simples, após carregar a rotina em L.M. na memória basta chamá-la pela função LSR("texto") ou LSR(var%). O posicionamento na tela é feito pelo comando PSET, PRESET ou outro comando gráfico e as cores de frente e fundo são dadas pelo comando COLOR.

1000 DATA 3A,63,F6,FE,03,C0,21,F8	0000
1010 DATA F7,71,23,66,6F,7E,32,E0	0001
1020 DATA D0,23,7E,32,E1,D0,23,7E	0002
1030 DATA 32,E1,D0,3A,B5,1C,32,DE	0003
1040 DATA D0,06,0B,C5,CD,2B,D0,C1	0004
1050 DATA 10,F7,19,0E,01,3A,E0,D0	0005
1060 DATA 32,F1,D0,C7,7A,B3,FC,32	0006
1070 DATA DD,D0,2A,F1,D0,7E,F7,26	0007
1080 DATA 00,6F,29,29,29,11,BF,1B	0008
1090 DATA 19,3E,0B,90,C7,4F,06,00	0009
1100 DATA 09,7E,06,06,17,CD,6F,D0	0010
1110 DATA 10,FA,11,E1,21,3A,E3,D0	0011
1120 DATA 3D,32,F3,D0,20,D7,21,DE	0012
1130 DATA D0,34,C1,0D,20,BF,C9,21	0013
1140 DATA DE,D0,36,FF,3B,07,36,00	0014
1150 DATA F5,C5,CD,BB,D0,21,DD,D0	0015
1160 DATA 34,CD,BB,D0,21,DD,D0,34	0016
1170 DATA 11,F1,C7,3A,DE,C0,4F,E6	0017
1180 DATA 07,6F,79,CB,3E,CB,3F,CB	0018
1190 DATA 3E,67,1A,DD,D0,E6,FB,06	0019
1200 DATA 00,4F,09,3A,DD,D0,E6,07	0020
1210 DATA 47,3E,0B,90,47,AF,37,17	0021
1220 DATA 10,ED,47,3A,DE,D0,A7,2B	0022
1230 DATA 06,CD,4A,00,80,1B,07,7B	0023
1240 DATA 2F,47,CD,4A,00,A0,CD,4D	0024
1250 DATA 00,3A,E9,F3,B7,B7,B7,B7	0025
1260 DATA 47,3A,FA,F3,B0,01,00,20	0026
1270 DATA 09,CD,4D,00,C9,FA,6A,00	0027
1280 DATA 14,11,B0,00,FIM	0028
1290 CLS:FOR I=&H0000 TO &H00E3:READ AS:DEF	0029
POKE I,VAL("&H"+AS):NEXT I:END	0030

O próximo programa é um exemplo de utilização da rotina,

```
10 SCREEN 2
20 PRESET (10,80):COLOR 11,4
30 DEFUSR=&HD000
40 AS-LSR("Milton Maldonado Jr.")
50 COLOR 15,1,1
60 GOTO 60
```

```
100
110
120
130
140
150
160
```

Dica especial para mudar a altura das letras use POKE &HD02C n com n variando de 1 a 8. Que acontece quando n vale 8?



EDITORAL ALEPH 1987  
EDITORAL ALEPH 1987  
EDITORAL ALEPH 1987  
EDITORAL ALEPH 1987  
EDITORAL ALEPH 1987  
EDITORAL ALEPH 1987  
EDITORAL ALEPH 1987  
EDITORAL ALEPH 1987



## DICAS SONORAS

Neste capítulo abordamos vários recursos sonoros disponíveis nos micros MSX

Além dos métodos normais, usando o PLAY e o SOUND, comentam-se alguns métodos não standard de geração de sons.

3.1 - Produzindo sons com ecos	87
3.2 - Percussão com o PLAY .. . . . .	88
3.3 - PLAY com SOUND .. . . . .	89
3.4 - Percussão com o click do teclado	90
3.5 - Teclado plano . . . . .	91
3.6 - Editor de sons .. . . . .	95
3.7 - Músicas com programas .. . . . .	97
3.8 - Musica aleatória .. . . . .	98
3.9 - Partitura sonora .. . . . .	100
3.A - Despertador . . . . .	102
3.B - O sintetizador de voz . . . . .	103

### 3.1 - PRODUZINDO SOMS COM ECOS

A produção de sons com ecos nos micros MSX é extremamente simples através do comando PLAY. Para isso basta gerar um mesmo som nos três canais, porém com um pequeno atraso um em relação aos outros.

Veja o programa a seguir.

```
10 PLAY "G0M5000"  
20 PLAY "CH32","R0CH32","R16CH32"
```

301  
888

Ele gera um som com eco.

Uma outra forma é usar um envelope periódico. Por exemplo, digite e rode o programa a listado a seguir.

```
10 PLAY "T24050M1000"  
20 PLAY "02CEGAA+AGI"  
30 PLAY "02CEGAA+AGF"  
40 PLAY "A03CDD+DC02A"  
50 PLAY "02CEGAA+AGF"  
60 PLAY "02DEFF+GFED"  
70 GOTO 20
```

2-8  
82  
102  
PCCH  
P478  
250  
1180

Agora atere a linha 10, deixando-a como mostramos abaixo e execute o programa novamente

```
10 PLAY "T2405BM1000"
```

### BIBLIOGRAFIA RECOMENDADA:

Linguagem Basic MSX - páginas 119 a 121  
Curso de Basic MSX v 1 - anexo 8  
Aprofundando-se no MSX - capítulo 5.

## 3.2 - PERCUSSÃO COM PLAY

A versatilidade do comando **PLAY** permite facilmente a geração de sons de percussão. Veja o exemplo a seguir:

```
10 PLAY "T120S0M9004L4AA06L16AAAAA4"  
20 GOTO 10
```



Note que usamos o envelope **0** com um período de modulação muito curto. Experimente agora substituir **T120** por **T140** e **M90** por **M200**.



## BIBLIOGRAFIA RECOMENDADA.

Linguagem Basic MSX - páginas 119 a 121  
Curso de Basic MSX v.1 - aula 8.  
Aprofundando-se no MSX - capítulo 5.

### 3.3 - PLAY COM SOUND

Uma outra forma de gerar sons de percussão com o comando PLAY é seleccionando ruído para um ou mais canais através de um comando SOUND

Experimente rodar o seguinte programa.

```
100 C$="02S0M1500LBT136CCM3500CM1500CCCM1500  
3500CM1500C  
110 SOUND 6,32  
120 FOR I=1 TO 2  
130   SOUND 7,28  
140   PLAY """, "", C$  
150 NEXT  
160 C$="S0M10000C1...."  
170 PLAY """, "", C$
```

10000  
11000  
12000  
13000  
14000  
15000  
16000  
17000

Altere o valor do registro B (sempre entre 1 e 32) e também a duração das notas e o tempo de execução. Você verá que pode produzir os sons mais variados. Um último recurso é mudar o formato de envoltória. Experimente alterá-lo, por exemplo para S13. Agora ao invés de uma "bateria", você deverá obter um "chocalho".



#### BIBLIOGRAFIA RECOMENDADA:

Linguagem Basic MSX - páginas 119 a 121 e 144 a 150  
Curso de Basic MSX v.1 - aula 8.  
Aprofundando-se no MSX - capítulo 5.



### 3.4 - PERCUSSÃO COM O CLICK DO TECLADO

Uma terceira e última forma de produzir sons de percussão no MSX é usando o click do teclado. Digite e execute o programa a seguir.

```
10 KEY(1)ON=KEY(2)ON
20 ON KEY GOSUB 70,80
30 XZ=1
40 OUT 170,127
50 OUT 170,255
60 FOR F% 1 1, XZ:NEXT F%:GOTO 40
70 XZ=XZ+1 : RETURN
80 XZ=XZ 1 ON XZ/0 : RETURN
```

801-  
810-  
828-  
831-  
847-  
859-  
866-  
877-

TOTAL = 2385

Use as teclas F1 e F2 para perceber o funcionamento do programa.

Esse recurso é usado de forma magistra no conhecido "ogurcho" PITALL. Além de música nos três canais ele usa o click do teclado para fazer a percussão.



### BIBLIOGRAFIA RECOMENDADA:

Coleção de Programas para MSX v 2 - página 15  
Aprofundando-se no MSX - página 87

### 3.5 - TECLADO PIANO

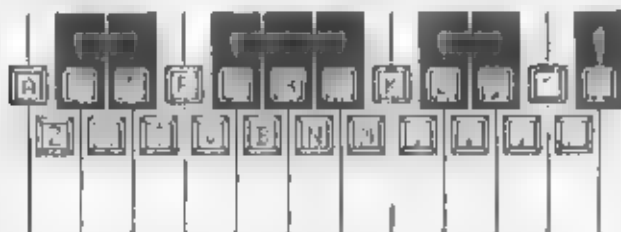
Existe uma maneira rápida e não muito sofisticada de se transformar um trecho do teclado de seu MSX num teclado musical.

No exemplo a seguir as duas filas inferiores de teclas do MSX serão utilizadas.



Estas teclas correspondem ao Expert 1.1. No Hotbit haverá uma pequena modificação que será comentada adiante.

Vamos atribuir a cada uma das letras (ou símbolos) dessas duas filas o papel de uma tecla branca ou preta, conforme o esquema a seguir:



Para isto basta digitar o programa listado adiante.

```

10 SCREEN0:WIDTH39:KEY OFF:POKE&HFCAB,1  616
20 DIM N(255):PLAY"S0M2000LB"             638
30 FOR I=0 TO 18                          1034
40 READ D$:D=ASC(D$):N(D)=I+41             657
50 NEXT I                                  1102
60 DATA Z,S,X,D,C,V,B,H,N,J,M,"","L,... 2848
C,.,/,*
70 A$=INPUT$(1):A=ASC(A$):N$="N"+STR$(N( 3008
A))
80 PRINT A$,:PRINT N$:PLAY N$              4128
90 GOTO 70                                 4207

```

TOTAL = 4207

Os que tiverem um Hotbit devem alterar a linha 60 para,

```
60 DATA Z,S,X,D,C,V,G,B,H,N,J,M,"",",",L,,  
6,/,>,[
```

Nesta linha são definidas as teclas que fazem, em sequência, o papel das teclas de um piano. Você pode alterar esta linha para usar outro trecho do teclado do MSX. Lembre-se, porém, de ajustar o valor final de I no laço que vai de 30 a 50.

Na linha 40 o comando N(D)=I+36 ajusta a escala musical de maneira a fazer a primeira tecla (no nosso caso a Z) tocar o Dó central (N36)

Você também pode ajustar este valor de maneira a pegar outros trechos da escala musical em seu teclado

Se você quiser alterar o timbre do seu "piano", modifique o PLAY da linha 20

Como alteração final, sugerimos alguns acréscimos nas linhas 20, 70 e 80 de maneira a ficarem assim:

```
10 SCREEN0:WIDTH39:KEY OFF:POKE&HFCAB,1 600  
20 DIM N(255):PLAY"50M2000L8","50M2000L8 600  
",,"50M2000L8"  
30 FOR I=0 TO 18 11:24  
40 READ D$:D=ASC(D$):N(D)=I+36 1:59A  
50 NEXT I 2:16  
60 DATA Z,S,X,D,C,V,G,B,H,N,J,M,"",",",L,, 289E  
C,1,/,*  
70 A$=INPUT$(1):A=ASC(A$):N$="N"+STR$(N( 3003  
A)):NAS="N"+STR$(N(A)+4):NBS="N"+STR$(N( 3003  
A)+7)  
80 PRINT A$,:PRINT N$:IF N(A)=0 THEN GOT 3003  
0 70 ELSE PLAY N$,NAS,NBS  
90 GOTO 70 3003
```

Experimente, ainda, alterar o comando SCREEN na linha 10 para: SCREEN0,,0

Você deve estar curioso para saber como foram geradas as figuras que ilustram o começo desta dica. Para fazer isto, basta digitar este complemento do programa:

```
1000 OPEN"GRP:"AS #1  
1010 COLOR 1,15,15:SCREEN 2  
1020 TS="C1S4R10E2D14H2L10G2U14F2D10G2R1  
4H2U10E2L14"
```

```

1030 FOR I= 1 TO 12
1040 READ A$
1050 PRESET(20*I,100):DRAW T$
1060 PRESET(20*I+3,102):PRINTN1,A$
1070 NEXT I
1080 DATA A,S,D,F,G,H,J,K,L,C,~,*
1090 FOR I= 1 TO 11
1100 READ A$
1110 PRESET(20*I+10,119):DRAW T$
1120 PRESET(20*I+13,121):PRINTN1,A$
1130 NEXT I
1140 DATA Z,X,C,V,B,N,H,"","'.,,5,/'
1150 T1$="C1L20U48R9U36L18D36R9"
1160 T2$="C1L20U48BU18U18"
1170 FOR I=0 TO 11
1180 PRESET (45+20*I,162):IF I=0 OR I=3
OR I=7 OR I=10 THEN DRAW T2$ ELSE DRAW T
1$:PAINT(25+20*I,90),1,1
1190 NEXT I
1200 RETURN

```

Além disso você deve alterar as linhas 10 e 80 do programa original para:

```

10 POKE &HFCAB,1
80 IF N(A)=0 THEN GOTO 70 ELSE PLAY N$,N
A$,NBS

```

e acrescentar a linha 88.

```

68 GOSUB 1000

```

Obviamente, quem tiver um Hotbit deve fazer as correspondentes alterações nas linhas 1080 e 1140.

Se você ainda quiser dar uma última sofisticada em seu programa, acrescente-lhe as linhas a seguir.

```

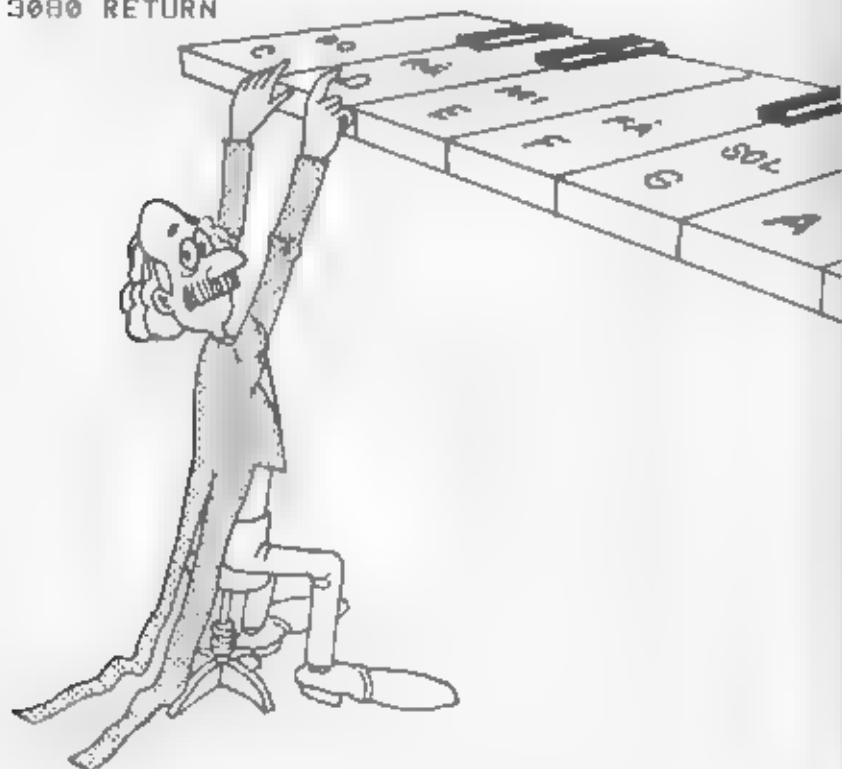
25 GOSUB 3000
80 IF N(A)=0 THEN GOTO 70 ELSE PLAY N$,N
A$,NBS:PUT SPRITE 0,(21+10*X(P),139-19*Y
(P)),0,1
1015 FOR S=1 TO 8
1016 IF S=8 THEN S$=S$+CHR$(127) ELSE S$
=S$+CHR$(65)

```

```

1017 NEXT S
1018 SPRITE$(1)=S%
3000 DIM X(19),Y(19)
3010 X(0)=1
3020 FOR P=0 TO 18
3030 R=P MOD 12
3040 X(P+1)=X(P)+1:IF R=4 OR R=11 THEN X
(P+1)=X(P)+2
3050 IF X(P) MOD 2=1 THEN Y(P)=1 ELSE Y(P)=
2
3070 NEXT P
3080 RETURN

```



#### BIBLIOGRAFIA RECOMENDADA:

Curso de Basic MSX v.1 - aula 8.  
 Linguagem Basic MSX páginas 119 a 121.

### 3.6 - EDITOR DE SONS

O programa a seguir é de grande utilidade na procura de valores corretos para serem usados com o comando SOUND.

Ao ser executado ele apresenta uma tabela com os números dos registros do PSG à esquerda. Para selecionar um dado registro deve-se usar a tecla de seta para cima. O número do registro selecionado aparecerá na parte inferior da tela logo abaixo da tabela. Para a terar seu valor deve-se usar as teclas de seta para a esquerda e seta para a direita. O registro 7 (de mixagem) é a terado de forma diferente, inicialmente deve ser selecionado com a tecla de seta para cima depois deve-se pressionar a barra de espaços a seguir basta pressionar 8 teclas 0 ou 1, para programar em binário o registro, após ter inserido os valores 0 e 1 nos bits do registro 7 pode-se pressionar RETURN para voltar ao modo de seleção normal do programa.

É recomendável que se conheça o funcionamento dos registros do PSG para programá-los conscientemente

```
10 *****
20 * BY THE DOCTOR LUZ *
30 *****
40 FOR L=&HC000 TO &HC145
50 READ A$:POKE L,VAL('8H'+A$)
60 NEXT
70 DEF USR=&HC000
80 PRINT"RODAR PROGRAMA ?"
90 A$=INPUT$(1)
100 IF A$="S" THEN A=USR(0) ELSE END
1000 DATA C0,C3,00,21,3C,C1,11,00
1010 DATA 00,01,0A,00,CD,7C,00,3E
1020 DATA 07,1E,38,CD,93,00,3E,08
1030 DATA 1E,0F,CD,93,00,3E,09,CD
1040 DATA 93,00,3E,0A,CD,93,00,26
1050 DATA 01,2E,02,CD,C6,00,CD,76
1060 DATA C0,CD,9F,00,FE,5A,C9,FE
1070 DATA 1C,28,0F,FE,1D,28,21,FE
1080 DATA 1E,20,65,FE,20,CA,C8,C0
1090 DATA 18,DD,DD,21,1B,C1,3A,2A
1100 DATA C1,16,00,5F,DD,19,DD,7E
1110 DATA 00,C6,01,DD,77,00,18,C7
1120 DATA DD,21,1B,C1,3A,2A,C1,16
1130 DATA 00,5F,DD,19,DD,7E,00,D6
1140 DATA 01,DD,77,00,18,B1,3A,2B
```

0000
0001
0002
0003
0004
0005
0006
0007
0008
0009
000A
000B
000C
000D
000E
000F
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
001A
001B
001C
001D
001E
001F
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
002A
002B
002C
002D
002E
002F
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
003A
003B
003C
003D
003E
003F
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
004A
004B
004C
004D
004E
004F
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
005A
005B
005C
005D
005E
005F
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
006A
006B
006C
006D
006E
006F
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
007A
007B
007C
007D
007E
007F
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
008A
008B
008C
008D
008E
008F
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
009A
009B
009C
009D
009E
009F
00A0
00A1
00A2
00A3
00A4
00A5
00A6
00A7
00A8
00A9
00AA
00AB
00AC
00AD
00AE
00AF
00B0
00B1
00B2
00B3
00B4
00B5
00B6
00B7
00B8
00B9
00BA
00BB
00BC
00BD
00BE
00BF
00C0
00C1
00C2
00C3
00C4
00C5
00C6
00C7
00C8
00C9
00CA
00CB
00CC
00CD
00CE
00CF
00D0
00D1
00D2
00D3
00D4
00D5
00D6
00D7
00D8
00D9
00DA
00DB
00DC
00DD
00DE
00DF
00E0
00E1
00E2
00E3
00E4
00E5
00E6
00E7
00E8
00E9
00EA
00EB
00EC
00ED
00EE
00EF
00F0
00F1
00F2
00F3
00F4
00F5
00F6
00F7
00F8
00F9
00FA
00FB
00FC
00FD
00FE
00FF

```

1150 DATA C1,DD,21,1B,C1,DD,77,07
1160 DATA 06,0F,0E,00,59,CD,F6,C0
1170 DATA 3E,20,CD,A2,00,DD,7E,00
1180 DATA 5F,79,DD,23,0C,CD,93,00
1190 DATA CD,F6,C0,3E,0A,CD,A2,00
1200 DATA 3E,0D,CD,A2,00,10,DD,C9
1210 DATA 3E,0A,CD,A2,00,3E,0D,CD
1220 DATA A2,00,3A,2A,C1,3C,FE,0E
1230 DATA D4,C5,C0,32,2A,C1,5F,CD
1240 DATA F6,C0,C3,27,C0,3E,00,C9
1250 DATA 21,09,09,CD,C6,00,3E,4D
1260 DATA CD,A2,00,3E,49,CD,A2,00
1270 DATA 3E,58,CD,A2,00,06,0B,CD
1280 DATA 9F,00,CD,A2,00,CB,22,FE
1290 DATA 30,28,02,CB,C2,10,F0,7A
1300 DATA 32,28,C1,C3,27,C0,7B,CB
1310 DATA 3F,CB,3F,CB,3F,CB,3F,21
1320 DATA 2C,C1,D5,5F,16,00,19,7E
1330 DATA CD,A2,00,D1,21,2C,C1,7B
1340 DATA E6,0F,5F,16,00,19,7E,CD
1350 DATA A2,00,C9,3D,C1,49,C1,FA
1360 DATA 29,C1,00,0F,0F,0F,00,00
1370 DATA 00,00,00,00,30,31,32,33
1380 DATA 34,35,36,37,38,39,41,42
1390 DATA 43,44,45,46,52,45,47,2E
1400 DATA 20,56,41,4C,4F,52

```

```

2000
2005
3002
3006
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099

```



## BIBLIOGRAFIA RECOMENDADA:

Linguagem Basic MSX páginas 144 a 150.  
 Aprofundando-se no MSX capítulo 5.  
 Linguagem de Máquina MSX - páginas 144 a 148.

### 3.7 - MÚSICAS COM PROGRAMAS

O programa abaixo ilustra um típico caso de multiprocessamento das máquinas MSX. Enquanto executa as tarefas de processamento normal de um micro, o MSX pode simultaneamente, executar músicas. Apenas a título de exemplo, usamos um processamento bem simples que consiste apenas em mostrar na região central da tela alguns números.

Note que a cada 1/54 segundos o processamento normal é interrompido para que o gerador de sons possa ser programado. Enquanto o gerador de sons executa a programação, o processamento normal continua.

```
100 SCREEN 0,,0:WIDTH 39:KEY OFF
110 INTERVAL ON : DEFINT F,G
120 DIM P$(34),Q$(34)
130 GOSUB 250:K=1
140 P$(1)="t20050m7000"
150 Q$(1)="t20050m7000"
160 PLAY P$(1),Q$(1)
170 ON INTERVAL 54 GOSUB 620
180 FOR F=1 TO 1000
190 LOCATE 8,10,0
200 PRINT USING"#####";F^2;K
210 FOR G=1 TO 100 : NEXT G
220 NEXT F
230 GOTO 180
240 '
250 P$(2)="r4o3":Q$(2)="r4o3"
260 P$(3)="a4o4c2d4":Q$(3)="r4L64ao4cea"
270 P$(4)="e4.f0e4":Q$(4)="o3ao4cea"
280 P$(5)="d2o3b4":Q$(5)="o3gbo4dg"
290 P$(6)="g4.a8b4":Q$(6)="o3gbo4dg"
300 P$(7)="o4c2o3a4":Q$(7)="o3ao4cea"
310 P$(8)="a4.g#8a4":Q$(8)="dfa05d"
320 P$(9)="b2g#4":Q$(9)="o4eg#bo5d"
330 '
340 P$(10)="e2a4":Q$(10)="o4eg#bo5d"
350 P$(11)="o4c2d4":Q$(11)="o3ao4cea"
360 P$(12)="e4.fBf4":Q$(12)="o3ao4cea"
370 P$(13)="d2o3b4":Q$(13)="o3gbo4dg"
380 P$(14)="g4.a8b4":Q$(14)="o3gbo4dg"
390 P$(15)="o4c4.o3b8a4":Q$(15)="o3ao4cea"
400 P$(16)="g#4.f#8g#4":Q$(16)="o4eg#bo5d"
410 P$(17)="a2.":Q$(17)="o3ao4cea"
420 P$(18) "a2r4":Q$(18)="o3ao4cea"
```



```

430 '
440 P$(19)~"o4g2." :Q$(19)="14o4ceg"
450 P$(20)~"g4.f8e4":Q$(20)~"ceg"
460 P$(21)~"d2o3b4":Q$(21)~"o3gbo4d"
470 P$(22)~"g4.a8b4":Q$(22)~"o3gbo4d"
480 P$(23)~"o4c2o3a4":Q$(23)~"o3ao4ce"
490 P$(24)~"a4.g#8a4":Q$(24)~"o3ao4ce"
500 P$(25)~"b2g#4":Q$(25)~"eg#b"
510 P$(26)~"e2.':Q$(26)~"eg#b"
520 '
530 P$(27)~"o4g2." :Q$(27)~"04CEG"
540 P$(28)~"g4.f8e4":Q$(28)~"ceg"
550 P$(29)~"d2o3b4":Q$(29)~"o3gbo4d"
560 P$(30)~"g4.a8b4":Q$(30)~"o3gbo4d"
570 P$(31)~"o4c4.o3b8a4":Q$(31)~"o3ao4ce"
580 P$(32)~"g#4.f#8g#4":Q$(32)~"eg#b"
590 P$(33)~"a2.a2R4":Q$(33)~"o3ao4ce o3a2"
600 RETURN
610 '
620 K=K+1
630 IF K=34 THEN K=2
640 PLAY P$(K),Q$(K)
650 RETURN

```



#### BIBLIOGRAFIA RECOMENDADA:

Linguagem Basic MSX - páginas 108 170 e 179

## 2.8 - MÚSICA ALEATÓRIA

Se você tiver curiosidade em saber como será a música do ano 3000, basta mandar seu MSX soltar a imaginação.

Digite o programa a seguir e ouça seu MSX "compondo". Quando você não aguentar mais digite CONTROL + STOP.

Se você, porém, tiver paciência e curiosidade ouça atentamente e verifique que há um certo padrão e algumas passagens geniais. Afinal a geração dos números aleatórios que estão sob a música é feita pelo micro segundo uma rig da regra matemática

```
100 PLAY "S0M8000","S0M8000","S0M8000"
110 L$="L"+STR$(INT(RND(TIME)*31)*2+2)
120 X$=L$+"N"+STR$(INT(RND(TIME)*60))
130 Y$=L$+"N"+STR$(INT(RND(TIME)*30+50))
140 Z$=L$+"N"+STR$(INT(RND(TIME)*16+80))
150 PLAY X$,Y$,Z$
160 GOTO 110
```

A composição do exemplo intitula-se "A ORDEM DO CAOS" e o andamento é "ALLEGRO VIVACE CON UN PIZZICO DI PAZZIA". Se você quiser alterar as regras mude a linha 100 (tentando outros envelopes) e as linhas 120 e 140, mudando a distribuição das notas. Lembre-se, porém, que o argumento do N no PLAY não pode ultrapassar 96.

Lembrando o famoso exemplo do físico JAMES JEANS (2 macacos imortais acorrentados a um plano por toda a eternidade acabarão tocando uma sonata de BEETHOVEN), tenha paciência e fique esperando a obra prima do seu MSX!



### 3.9 - PARTITURA SONORA

Executando-se alguns raros privilegiados que têm o chamado "ouvido absoluto", a maioria das pessoas tem uma certa dificuldade em identificar uma nota musical tocada individualmente.

Associar o som da nota a uma tecla do piano ou a uma posição na partitura musical se torna ainda mais difícil.

Para treinar seu ouvido e sua leitura de partitura digite o programa a seguir. Para fazer a nota "subir" ou "descer" pela escala musical basta usar as teclas de seta para cima e seta para baixo. Uma vez escolhida a nota, basta apertar a barra de espaços para ouvir seu som.

```

100 P$(1)="DO      C"=P$(2)="RE      D"
110 P$(3)="MI      E"=P$(4)="FA      F"
120 P$(5)="SO      G"=P$(6)="LA      A"
130 P$(7)="SI      B"
140 COLOR 15,1,1:SCREEN 2,,0
150 OPEN"GRP:"AS #1
160 PRESET(31,1/0)
170 PRINT#1,"OITAVA  NOTA  CIFRA"
180 FOR C=0 TO 1
190 FOR L=63+48*C TO 95+48*C STEP 8
200 LINE(0,L)-(255,L)
210 NEXT L,C
220 FOR I=1 TO 26
230 D=23+8*I
240 FOR L1(0,15)=4*I:PRINT#1,CHR$(1)+
    CHR$(73)
250 LINE(0,5)=(D/6,21),,D
260 IF IMOD D/3 OR IMOD D/6 THEN 210
270 LINE(0,3,5)=(D/3,15),,D
280 NEXT I
290 LINE(0,141)=(38,151):LINE(126,103)
    (1,4,10):LINE(27,55) (27,55)
300 D1$="C1512LHJRM+1,+1M+1,+2M 1,+2M
    +1M+,+1,+2M+1,2E4D2HJ166H"
310 PRESET(14,86):DRAW D1$
320 PRESET(14,87):DRAW D1$
330 D2$="C1512LHJERM+2,+1M+1,+2M 1,+3G2
340 PRESET(14,120):DRAW D2$
350 PRESET(14,121):DRAW D2$:PSET(27,115)
:GOTO 2,123)

```

360 FOR T=1 TO 8:CS=CS+CHR\$(PEEK(7222+T))	360
:DS=DS+CHR\$(PEEK(7206+T)):NEXT T:SPRITE	
%1)=CS:SPRITE%(2)-DS	
370 I=13:PLAY"50M7000"	
380 A=STICK(0)	380
390 I-I+(A-5)-(A=1):X=23+8*I:Y=152-4*I	390
400 IF I<1 THEN I=1	400
410 IF I>26 THEN I=26	410
420 O=(I+15)\7:W=(I+15)MOD7+1	420
430 IF A=0 THEN 460	430
440 LINE (40,180)-(240,190),1,BF	440
450 PRESET(41,181):PRINTh1,0;" ";	450
W)	
460 PUT SPRITE 3,(X,Y-1),,1	460
470 PUT SPRITE 4,(X,22),,2	470
480 N=12*(I\7)+14+2*(IMOD7)+(IMOD7*1)+(I	480
MOD7*5)	
490 AS="N"+STR\$(N)	490
500 IF STRIG(0) THEN GOSUB 520	500
510 GOTO 380	510
520 PLAY AS	520
530 IF STRIG(0) THEN 530	530
540 RETURN	540



### 3.A - DESPERTADOR

Se você quiser usar seu MSX como relógio despertador, digite o programa a seguir e rode-o. Forneça a hora, minuto e segundo do momento de despertar e depois atualize o horário. Ao digitar o valor do SEGUNDO ATUAL digite um número um pouco maior e fique esperando até seu relógio de pulso indicar o mesmo número. Nesse instante pressione RETURN.

100 SCREEN 1:D 0:KEY OFF	94F
110 INPUT HORA DO DESPERTAR":HD	DC6
120 INPUT MINUTO DO DESPERTAR":MD	16C8
130 INPUT "SEGUNDO DO DESPERTAR":SD	EC9D
140 INPUT HORA ATUAL":H0	7474
150 INPUT MINUTO ATUAL":M0	5270
160 INPUT "SEGUNDO ATUAL":S0	1038
170 TIME 0	C144
180 LOCATE 3,10,0:PRINT "DESPERTADOR PARA "	0658
190 PRINT USING "##:##:##":HD;MD;SD	7822
200 LOCATE 3,1:PRINT "HORA CLRIA "	8422
210 TIME\60:S1 (C011)MOD 60	8174
220 M=(011)\60:M1 (M011) MOD 60	8198
230 H=(M+M0)\60:H1 (M011) MOD 24	81F5
240 PRINT USING "##:##:##":H1;M1;S1	88F0
250 D (H1 HD)+(M1 MD)+(S1 SD)	D48
260 IF D > 3 THEN PLAY "V1EN40N4:"	0958
270 IF STRIG(0) THEN D=0	0F57
280 GOTO 200	D497

TOTAL = 0591

Não esqueça de deixar o volume do micro ou da TV no máximo e siga o seguinte procedimento: vá dormir. Ao tocar o alarme acorde! Levante e rastejando aproxime-se do micro. Pressione a barra de espaços para acabar com o maldito barulho e não volte para a cama! Se quiser um barulho mais irritante atore o PLAY da linha 260. Se chegar atrasado no serviço, leve a listagem do programa para seu chefe e tente descobrir juntos, onde você errou na digitação!

### BIBLIOGRAFIA RECOMENDADA:

LINGUAGEM BASIC MSX página 166

### 3 0 - DIGITALIZADOR DE VOZ

Com este programa você pode digitalizar qualquer som e depois reproduzi-lo através do canal de áudio do micro. É muito útil em entradas de programas, que podem ser o trecho de uma música ou mesmo sua própria voz.

A entrada do som é feita pelo cabo de entrada (EAR) do cassete, e o som reproduzido sai pelo canal de áudio. Se o seu micro é um Expert o som sairá pelo alto-falante interno. No Hot-Bit este sairá pela televisão.

A duração do som gravado é determinada pelos endereços &HC007 e &HC035. Os valores destes dois bytes devem ser iguais e compreendidos entre 0 e 170 e a maior duração é obtida com o valor 0. Para mudar a duração basta dar POKES nestes endereços com os valores acima especificados.

Você pode salvar uma informação digitada em disco ou fita com o comando:

```
BSAVE "NOME",C1,&HB000
```

onde

```
ST=258*PEEK(&HC007)
```

Naturalmente se o valor dado a &HC007 for menor que 128 o BSAVE não terá o efeito desejado pois este salvará uma parte da ROM em vez de salvar a RAM abaixo de &H0000.

```
1000 REM -
1010 REM Digitalizador de Voz
1020 REM By The Pilot 1988
1030 REM -----
1040 DATA F3,3F,AA,D3,AB,1,00,80
1050 DATA 11,00,80,AF,06,03,4F,DB
1060 DATA A2,F6,80,B1,CD,0,00,00
1070 DATA 00,00,00,00,00,00,10,EE
1080 DATA 77,23,E5,ED,52,14,0D,E4
1090 DATA 4E,00,D3,AB,FB,C7,F3,3E
1100 DATA 00,D3,AB,21,00,80,11,00
1110 DATA 80,7E,06,08,4F,1F,1F,D1
1120 DATA AA,79,E3,3F,00,00,00,00
1130 DATA 00,00,00,00,10,EE,23,E5
1140 DATA ED,52,E1,38,E4,C7,AB,D3
1150 DATA AB,FB,C9,FIM
```

```
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```

1160 GOTO FOR I=&HC000 TO &HC05A:READ A$:
POKE I,VAL("&H"+A$):NEXT I
1170 PRINT "GRAVAR:DLFUSR=&HC000:PRINT U
GR(0)
1180 PRINT "TOCAR: DEFPCR=&HC02E:PRINT U
GR(0)
1190 END

```

Se o seu micro for um Hotbit mude a linha 1180 para:

```

1190 POKE &HC002,255:POKE &HC030,255:END

```



#### BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulo 3



## DICAS PARA O CASSETE

Neste capítulo abordaremos o gravador cassete  
seus recursos e suas utilizações

Além dos recursos disponíveis normalmente no  
BASIC várias aplicações somente possíveis através da  
Linguagem de Máquina são abordadas. O gravador pode  
ser usado com mais segurança e maior rapidez

4.1	- Auto-execução de programas em cassete	106
4.2	- Gravando programas BASIC em binário ...	106
4.3	- Gravando textos . . . . .	112
4.4	- Lendo textos ..... ..	113
4.5	- Verificando gravações .... .	114
4.6	- Gravador de SCREEN 2 ..... .	116



#### 4.1 - AUTO-EXECUÇÃO DE PROGRAMAS EM CASSETE

O programa apresentado a seguir gera e grava em fita uma pequena rotina em linguagem de Máquina capaz de carregar e auto executar programas em BASIC gravados por CSAVE.

O procedimento para usá-lo é o seguinte:

- ▶ digite e execute o programa
- ▶ prepare o gravador cassete com a fita em que será gravado o programa em BASIC
- ▶ digite RETJRN para gravar a rotina em L.M (ela será gravada com o nome AJTOMA)
- ▶ digite NEW
- ▶ digite o programa em BASIC a ser gravado em fita cassete.
- ▶ grave-o com o comando.

"ONLY MATTER X"

Agora a fita deve conter, em sequência, o programa AUTOMA (gravado por BSAVE) e o programa MATRIX (gravado por CSAVE). O programa AUTOMA pode ter qualquer outro nome porém o programa MATRIX deve ser gravado exatamente com esse nome (com as letras maiúsculas).

Para carregar e auto-executar o programa MATRIX basta posicionar a fita no programa AJTOMA e comandar.

BLOAD"CASE="R

Quando for carregado para a memória do micro o programa AUTOMA carregará e executará o programa MATRIX1.

## PROGRAMA GERADOR DO AUTOMA

100 REM	100
110 REM AUTOCAS.BAS	110
120 REM	120
130 SCREEN 0	130
140 WIDTH 39	140
150 CLEAR 200,&HC000	150
160 FOR F=&HC000 TO &HC041	160
170 READ A\$	170
180 A=VAL("&H"+A\$)	180
190 POKE F,A	190
200 NEXT F	200
210 PRINT,,,, "PREPARE O GRAVADOR E ";	210

```

220 PRINT "PRESSIONE RETURN!"
230 AS=INPUT$(1)
240 BSAVE"CAS:AUTOMA",&HC000,&HC041
250 END
260 REM
270 REM DATA's para L.H.
280 REM
290 DATA F3,FD,21,B0,FB,FD,36,00
300 DATA 00,FD,36,01,01,21,34,C0
310 DATA 11,F0,F8,01,0D,00,ED,80
320 DATA 21,F0,F8,22,FA,F3,11,0E
330 DATA 00,19,22,F8,F3,21,2B,C0
340 DATA CD,3F,70,22,4D,41,54,52
350 DATA 49,58,22,00,43,4F,4C,4F
360 DATA 52,37,2C,31,3A,52,55,4E
370 DATA 0D,49,00,00,00,00,00,00

```

```

370
380
390
400
410
420
430
440
450
460
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```



## BIBLIOGRAFIA

Aprofundando-se no MSX - capítulo 8.  
 Programação Avançada em MSX - capítulo 4

## 4.2 - GRAVANDO PROGRAMAS BASIC EM BINÁRIO

Quando dispomos apenas do gravador cassete para armazenar dados de um programa o processo de gravação é lento e torna-se pouco confiável. Uma forma de torná-lo mais prático é armazenando o programa e seus dados conjuntamente em formato binário. Para isso devemos dispor de um pequeno programa em Linguagem de Máquina que deve preparar a memória do micro para poder gravar, e sem perder os dados.

Vamos exemplificar o que acabamos de descrever. Imagine que você queira armazenar a matriz gerada pelo programa da figura a seguir:

```
100 DIM A$(5000,1)
120 FOR F=0 TO 5000
130   FOR G=0 TO 1
140     A$(F,G)=STR$(2*F+G^2)
150   NEXT G
160 NEXT F
```

200  
232  
244  
242  
268  
280

Uma das maneiras de fazer isso seria abrir um arquivo no cassete e gravar cada um dos dados nesse arquivo. Esse processo é extremamente demorado e portanto mais sujeito a falhas. Se pudermos gravar em binário (com BSAVE) a região da memória que contém as variáveis, e até o próprio programa, economizaremos tempo e aumentaremos a confiabilidade dos dados.

A seguir vamos apresentar um programa em BASIC que ao ser executado, gera em fita ou em disco um programa em Linguagem de Máquina com o nome "BASBIN" (em fita) ou "BASBIN.BIN" (em disco). Antes de prosseguir, digite-o e execute-o. Depois, verifique se o programa BASBIN foi realmente gerado.

```
100 REM
110 REM BINARIZADOR DE BASIC
120 REM
130 E=&HD000
140 READ A$ : A=VAL("&H"+A$)
150 IF A$="XX" THEN 170
160 POKE E,A : E=E+1 : GOTO 140
170 E=&HD100
180 READ A$ : A=VAL("&H"+A$)
190 IF A$="XX" THEN 210
200 POKE E,A : E=E+1 : GOTO 180
210 E=&HD200
```

100  
100  
110  
154  
160  
1400  
200  
282  
318  
386  
386  
386

```

220 READ A$: 1 A=VAL("&H"+A$)
230 IF A$="XX" THEN 250
240 POKE E,A : E=E+1 : GOTO 220
250 BSAVE "BASBIN.BIN",&H0000,&H28C
260 END
270 RLM DADOS EM &H0000
280 DATA F3,21,07,80,ED,48,80,F3
290 DATA CD,4C,D0,ED,48,77,F1,CD
300 DATA 4C,D0,ED,48,6A,F1,CD,4C
310 DATA D0,ED,48,68,F1,CD,4C,D0
320 DATA ED,48,68,F1,CD,4C,D0,ED
330 DATA 48,A0,F0,CD,4C,D0,ED,48
340 DATA C6,F6,CD,4C,D0,ED,48,C4
350 DATA F6,CD,4C,D0,ED,48,C2,F6
360 DATA CD,4C,D0,ED,48,76,F6,CD
370 DATA 4C,D0,FB,C9,71,23,70,23
380 DATA C9,48,50,45,4C,41,20,41
390 DATA 52,4F,54,49,44,45,4F,54
400 DATA 41,4E,45,52,XX
410 RLM DADOS EM &H100
420 DATA F3,21,07,80,01,80,F3,CD
430 DATA 4D,D1,01,77,F1,CD,4D,01
440 DATA 01,6A,F1,CD,4D,D1,01,68
450 DATA F1,CD,4D,D1,01,68,F1,CD
460 DATA 4D,D1,01,A0,F0,CD,4D,D1
470 DATA 01,C6,F6,CD,4D,D1,01,C4
480 DATA F6,CD,4D,D1,01,C2,F6,CD
490 DATA 4D,D1,01,76,F6,CD,4D,D1
500 DATA 21,55,D1,11,07,80,01,14
510 DATA 00,ED,80,FB,C9,71,02,23
520 DATA 03,7F,02,23,C9,45,44,49
530 DATA 54,4F,52,41,20,41,4C,45
540 DATA 50,48,20,2D,20,31,39,3B
550 DATA 37,XX
560 RLM DADOS EM &H200
570 DATA F3,21,07,80,01,80,F3,CD
580 DATA 65,D2,01,77,F1,CD,65,D2
590 DATA 01,6A,F1,CD,65,D2,01,68
600 DATA F1,CD,65,D2,01,68,F1,CD
610 DATA 65,D2,01,A0,F0,CD,65,D2
620 DATA 01,C6,F6,CD,65,D2,01,C4
630 DATA F6,CD,65,D2,01,C2,F6,CD
640 DATA 65,D2,01,76,F6,CD,65,D2
650 DATA 21,7B,D2,11,07,80,01,14
660 DATA 00,ED,80,21,6D,D2,11,F0
670 DATA FB,01,08,00,ED,80,21,F0
680 DATA FB,22,FA,F3,11,0C,00,19
690 DATA 22,FB,F3,FB,C9,7E,02,23
700 DATA 03,7E,02,23,C9,47,4F,54

```

```

710 DATA 4F,20,30,30,31,32,30,0D
720 DATA 45,44,49,54,4F,52,41,20
730 DATA 41,4C,45,50,48,20,2D,20
740 DATA 31,39,38,37,XX

```

```

1500
1510
1520
1530

```

Uma vez com o programa BASBIN gravado, vamos testá-lo ao mesmo tempo em que aproveitamos para aprender como ele deve ser usado. Para isso limpe a memória com um NEW e digite o programa a seguir exatamente como ele está listado, sem nenhum espaço a menos ou a mais!

```

100 REM EDITORA ALEPH      1987
110 CLEAR 200,&HD000
120 X$="EDITORA ALEPH - 1987"
130 PLAY"t240S8M1000"
140 PLAY"02CEGAA+AGE"
150 PLAY"02CEGAA+AGE"
160 PLAY"FA03C00+DC02A"
170 PLAY"02CEGAA+AGE"
180 PLAY"02DEFF+GFED"
190 SCREEN 0:PRINT "X$"
200 GOTO 140

```

```

1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650

```

10000 = 10000

Após tê-lo digitado, execute-o espere alguns segundos e pressione CONTROL+STOP para interrompê-lo. Vamos agora preparar a memória para poder ser gravada em binário. Digite o seguinte comando:

```
BLOAD"BASBIN.BIN",R
```

A seguir, vamos gravar a memória com o programa e com os dados. Para isso comande:

```
BSAVE"TESTES.BIN",&H8000,&HD300,&HD200
```

Com isso o programa em BASIC e seus dados serão gravados em formato binário.

Para tornar a carregá-lo, basta comandar:

```
BLOAD"TESTES.BIN
```

E a seguir, comander

```
DEFUSR=&HD100:USR(0)
```

Experimente fazer isso e depois comande:

```
PRINT X$
```

Você verá que o conteúdo de X\$ ainda está presente.

Se tivéssemos comandado,

```
BLOAD"BASBIN.BIN",R
```

O programa seria carregado e começaria a rodar automaticamente a partir da linha 120. Essa é uma outra forma de fazer programas em BASIC se auto-executarem logo após a carga a partir de fitas cassete.

Você pode usar o BASBIN.BIN com qualquer programa em BASIC, desde que ele não ocupe a memória acima de &H0000. O programa em BASIC deverá começar sempre com as duas primeiras como mostradas a seguir:

```
100 REM EDITORA ALEPH 1937  
110 CLEAR 200,&H0000
```

A rigor, o primeiro parâmetro do CLEAR da linha 110 pode ser alterado, mas o segundo deve ser necessariamente menor ou igual a &H0000 !



#### BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulo 1  
Programação Avançada em MSX - capítulo 1

### 4.3 -GRAVANDO TEXTOS

Este programinha transfere uma mensagem (em ASCII) para a fita cassete. Após a chamada, ele transfere para a fita os dados definidos após &HC016 até o primeiro byte &H00, retornando então ao interpretador BASIC.

```
1000 DATA CD,EA,00,21,16,C0,7E,A7
1010 DATA 28,08,E5,CD,ED,00,E1,23
1020 DATA 18,F4,CD,F0,00,C9
1025 REM a seguir esta a mensagem
1030 DATA 4F,53,20,43,59,4C,4F,4E
1040 DATA 49,4F,53,20,49,4E,56,41
1050 DATA 44,49,52,41,4F,20,41,20
1060 DATA 54,45,52,52,41,20,21,00
1070 CLS:FOR I=&HC000 TO &HC035
1080 READ AS:POKE I,VAL("&H"+AS):NEXT I
END
```

003
01E
1230
1F2H
24F6
3704
53F9
55FA
6277
1083



### BIBLIOGRAFIA RECOMENDADA.

Programação Avançada em MSX - capítulo 4.

#### 4.4 - LENDU TEXTOS

Este programa faz a função oposta à da dica 4.3, lê a fita e transfere para a tela a informação lida. Devido à simplicidade deste sistema, caracteres de controle (CR, LF, etc) não são reconhecidos como tais e aparecem na tela como símbolos gráficos. A leitura termina na primeira ocorrência do byte &H00.

Para posicionar a informação na tela use os comandos

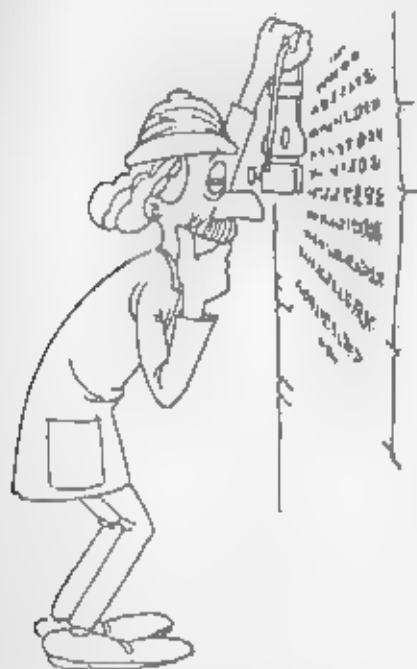
LOCATE C.L. PRINT ;

onde C e L são as coordenadas de impressão desejada.

```

1000 DATA C0,F1,00,CD,E4,00,A7,28
1010 DATA 04,D3,98,18,F6,CD,E7,00
1020 DATA C9,FIM
1030 CLS:FOR I=&HC100 TO &HC110:READ A5:
POKE I,VAL("&H"+A5):NEXT I:END

```



**10-061**

Programação Avançada em MSX - capítulo 4



## 4.5 - VERIFICANDO GRAVAÇÕES

Quantas vezes você já não deu socos no seu gravador por causa de um programa que não carregou? Para programas em BASIC ainda existe o comando "LOAD" mas para programas em binário nada foi implementado no BASIC MSX usando a verificação de gravações.

O programa a seguir insira a na RAM uma sub-rotina em L M que faz a função de verificar um arquivo gravado no formato binário. Eis o programa.

```

1000 REM -----
1010 REM VERIFICADOR DE BSAVE V. 1.0
1020 REM (C) 1987/88 BY THE PILOT
1030 REM -----
1040 DATA 1D,6E,00,0F,D3,77,D3,99
1050 DATA 1D,E1,00,08,76,00,0A,C5
1060 DATA 1D,E4,00,C1,10,F7,FE,D0
1070 DATA 10,1F,06,06,C5,C0,E4,00
1080 DATA 10,71,D1,70,C1,10,15,3E
1090 DATA 10,1D,71,01,70,C0,E1,00
1100 DATA 21,77,11,06,06,E5,C5,CD
1110 DATA 14,00,11,E1,77,23,10,F5
1120 DATA 3E,71,1A,57,18,1D,09,D0
1130 DATA D3,90,2A,79,10,1D,09,D0
1140 DATA D3,70,2A,58,18,1D,09,D0
1150 DATA 3E,70,06,12,D7,90,10,11
1160 DATA 2A,77,18,ED,70,79,18,13
1170 DATA 0F,77,5D,18,E5,D5,CD,E4
1180 DATA 00,C1,11,38,0E,8E,C4,A2
1190 DATA D0,11,07,E5,ED,52,E1,20
1200 DATA 1B,1D,05,3E,01,32,5D,F8
1210 DATA 21,10,00,1D,C6,00,1D,E7
1220 DATA 00,7A,7D,F8,A7,C0,71,11
1230 DATA D0,7E,A7,C8,23,C0,A7,00
1240 DATA 18,F7,CD,A9,D0,CD,84,D0
1250 DATA 19,F5,7C,CD,C7,00,7D,CD
1260 DATA C7,D0,11,C9,F5,3E,2D,D3
1270 DATA 90,11,C0,C7,D0,3E,2D,D3
1280 DATA 90,11,01,32,5D,F3,C9,F5
1290 DATA 18,7F,10,3F,1B,3F,C8,3F
1300 DATA 01,11,D0,01,4F,3E,00,88
1310 DATA 47,0A,D3,90,11,16,0F,01
1320 DATA EC,C0,81,4F,3F,00,81,47
1330 DATA 0A,D3,98,C9,70,11,32,33
1340 DATA 34,3C,36,37,30,39,41,42
1350 DATA 43,44,45,46,0D,0A,53,45
1360 DATA 4D,20,45,52,52,4F,53,0D

```

0000  
 0001  
 0002  
 0003  
 0004  
 0005  
 0006  
 0007  
 0008  
 0009  
 0010  
 0011  
 0012  
 0013  
 0014  
 0015  
 0016  
 0017  
 0018  
 0019  
 0020  
 0021  
 0022  
 0023  
 0024  
 0025  
 0026  
 0027  
 0028  
 0029  
 0030  
 0031  
 0032  
 0033  
 0034  
 0035  
 0036  
 0037  
 0038  
 0039  
 0040  
 0041  
 0042  
 0043  
 0044  
 0045  
 0046  
 0047  
 0048  
 0049  
 0050  
 0051  
 0052  
 0053  
 0054  
 0055  
 0056  
 0057  
 0058  
 0059  
 0060  
 0061  
 0062  
 0063  
 0064  
 0065  
 0066  
 0067  
 0068  
 0069  
 0070  
 0071  
 0072  
 0073  
 0074  
 0075  
 0076  
 0077  
 0078  
 0079  
 0080  
 0081  
 0082  
 0083  
 0084  
 0085  
 0086  
 0087  
 0088  
 0089  
 0090  
 0091  
 0092  
 0093  
 0094  
 0095  
 0096  
 0097  
 0098  
 0099  
 0100  
 0101  
 0102  
 0103  
 0104  
 0105  
 0106  
 0107  
 0108  
 0109  
 0110  
 0111  
 0112  
 0113  
 0114  
 0115  
 0116  
 0117  
 0118  
 0119  
 0120  
 0121  
 0122  
 0123  
 0124  
 0125  
 0126  
 0127  
 0128  
 0129  
 0130  
 0131  
 0132  
 0133  
 0134  
 0135  
 0136  
 0137  
 0138  
 0139  
 0140  
 0141  
 0142  
 0143  
 0144  
 0145  
 0146  
 0147  
 0148  
 0149  
 0150  
 0151  
 0152  
 0153  
 0154  
 0155  
 0156  
 0157  
 0158  
 0159  
 0160  
 0161  
 0162  
 0163  
 0164  
 0165  
 0166  
 0167  
 0168  
 0169  
 0170  
 0171  
 0172  
 0173  
 0174  
 0175  
 0176  
 0177  
 0178  
 0179  
 0180  
 0181  
 0182  
 0183  
 0184  
 0185  
 0186  
 0187  
 0188  
 0189  
 0190  
 0191  
 0192  
 0193  
 0194  
 0195  
 0196  
 0197  
 0198  
 0199  
 0200  
 0201  
 0202  
 0203  
 0204  
 0205  
 0206  
 0207  
 0208  
 0209  
 0210  
 0211  
 0212  
 0213  
 0214  
 0215  
 0216  
 0217  
 0218  
 0219  
 0220  
 0221  
 0222  
 0223  
 0224  
 0225  
 0226  
 0227  
 0228  
 0229  
 0230  
 0231  
 0232  
 0233  
 0234  
 0235  
 0236  
 0237  
 0238  
 0239  
 0240  
 0241  
 0242  
 0243  
 0244  
 0245  
 0246  
 0247  
 0248  
 0249  
 0250  
 0251  
 0252  
 0253  
 0254  
 0255  
 0256  
 0257  
 0258  
 0259  
 0260  
 0261  
 0262  
 0263  
 0264  
 0265  
 0266  
 0267  
 0268  
 0269  
 0270  
 0271  
 0272  
 0273  
 0274  
 0275  
 0276  
 0277  
 0278  
 0279  
 0280  
 0281  
 0282  
 0283  
 0284  
 0285  
 0286  
 0287  
 0288  
 0289  
 0290  
 0291  
 0292  
 0293  
 0294  
 0295  
 0296  
 0297  
 0298  
 0299  
 0300  
 0301  
 0302  
 0303  
 0304  
 0305  
 0306  
 0307  
 0308  
 0309  
 0310  
 0311  
 0312  
 0313  
 0314  
 0315  
 0316  
 0317  
 0318  
 0319  
 0320  
 0321  
 0322  
 0323  
 0324  
 0325  
 0326  
 0327  
 0328  
 0329  
 0330  
 0331  
 0332  
 0333  
 0334  
 0335  
 0336  
 0337  
 0338  
 0339  
 0340  
 0341  
 0342  
 0343  
 0344  
 0345  
 0346  
 0347  
 0348  
 0349  
 0350  
 0351  
 0352  
 0353  
 0354  
 0355  
 0356  
 0357  
 0358  
 0359  
 0360  
 0361  
 0362  
 0363  
 0364  
 0365  
 0366  
 0367  
 0368  
 0369  
 0370  
 0371  
 0372  
 0373  
 0374  
 0375  
 0376  
 0377  
 0378  
 0379  
 0380  
 0381  
 0382  
 0383  
 0384  
 0385  
 0386  
 0387  
 0388  
 0389  
 0390  
 0391  
 0392  
 0393  
 0394  
 0395  
 0396  
 0397  
 0398  
 0399  
 0400  
 0401  
 0402  
 0403  
 0404  
 0405  
 0406  
 0407  
 0408  
 0409  
 0410  
 0411  
 0412  
 0413  
 0414  
 0415  
 0416  
 0417  
 0418  
 0419  
 0420  
 0421  
 0422  
 0423  
 0424  
 0425  
 0426  
 0427  
 0428  
 0429  
 0430  
 0431  
 0432  
 0433  
 0434  
 0435  
 0436  
 0437  
 0438  
 0439  
 0440  
 0441  
 0442  
 0443  
 0444  
 0445  
 0446  
 0447  
 0448  
 0449  
 0450  
 0451  
 0452  
 0453  
 0454  
 0455  
 0456  
 0457  
 0458  
 0459  
 0460  
 0461  
 0462  
 0463  
 0464  
 0465  
 0466  
 0467  
 0468  
 0469  
 0470  
 0471  
 0472  
 0473  
 0474  
 0475  
 0476  
 0477  
 0478  
 0479  
 0480  
 0481  
 0482  
 0483  
 0484  
 0485  
 0486  
 0487  
 0488  
 0489  
 0490  
 0491  
 0492  
 0493  
 0494  
 0495  
 0496  
 0497  
 0498  
 0499  
 0500  
 0501  
 0502  
 0503  
 0504  
 0505  
 0506  
 0507  
 0508  
 0509  
 0510  
 0511  
 0512  
 0513  
 0514  
 0515  
 0516  
 0517  
 0518  
 0519  
 0520  
 0521  
 0522  
 0523  
 0524  
 0525  
 0526  
 0527  
 0528  
 0529  
 0530  
 0531  
 0532  
 0533  
 0534  
 0535  
 0536  
 0537  
 0538  
 0539  
 0540  
 0541  
 0542  
 0543  
 0544  
 0545  
 0546  
 0547  
 0548  
 0549  
 0550  
 0551  
 0552  
 0553  
 0554  
 0555  
 0556  
 0557  
 0558  
 0559  
 0560  
 0561  
 0562  
 0563  
 0564  
 0565  
 0566  
 0567  
 0568  
 0569  
 0570  
 0571  
 0572  
 0573  
 0574  
 0575  
 0576  
 0577  
 0578  
 0579  
 0580  
 0581  
 0582  
 0583  
 0584  
 0585  
 0586  
 0587  
 0588  
 0589  
 0590  
 0591  
 0592  
 0593  
 0594  
 0595  
 0596  
 0597  
 0598  
 0599  
 0600  
 0601  
 0602  
 0603  
 0604  
 0605  
 0606  
 0607  
 0608  
 0609  
 0610  
 0611  
 0612  
 0613  
 0614  
 0615  
 0616  
 0617  
 0618  
 0619  
 0620  
 0621  
 0622  
 0623  
 0624  
 0625  
 0626  
 0627  
 0628  
 0629  
 0630  
 0631  
 0632  
 0633  
 0634  
 0635  
 0636  
 0637  
 0638  
 0639  
 0640  
 0641  
 0642  
 0643  
 0644  
 0645  
 0646  
 0647  
 0648  
 0649  
 0650  
 0651  
 0652  
 0653  
 0654  
 0655  
 0656  
 0657  
 0658  
 0659  
 0660  
 0661  
 0662  
 0663  
 0664  
 0665  
 0666  
 0667  
 0668  
 0669  
 0670  
 0671  
 0672  
 0673  
 0674  
 0675  
 0676  
 0677  
 0678  
 0679  
 0680  
 0681  
 0682  
 0683  
 0684  
 0685  
 0686  
 0687  
 0688  
 0689  
 0690  
 0691  
 0692  
 0693  
 0694  
 0695  
 0696  
 0697  
 0698  
 0699  
 0700  
 0701  
 0702  
 0703  
 0704  
 0705  
 0706  
 0707  
 0708  
 0709  
 0710  
 0711  
 0712  
 0713  
 0714  
 0715  
 0716  
 0717  
 0718  
 0719  
 0720  
 0721  
 0722  
 0723  
 0724  
 0725  
 0726  
 0727  
 0728  
 0729  
 0730  
 0731  
 0732  
 0733  
 0734  
 0735  
 0736  
 0737  
 0738  
 0739  
 0740  
 0741  
 0742  
 0743  
 0744  
 0745  
 0746  
 0747  
 0748  
 0749  
 0750  
 0751  
 0752  
 0753  
 0754  
 0755  
 0756  
 0757  
 0758  
 0759  
 0760  
 0761  
 0762  
 0763  
 0764  
 0765  
 0766  
 0767  
 0768  
 0769  
 0770  
 0771  
 0772  
 0773  
 0774  
 0775  
 0776  
 0777  
 0778  
 0779  
 0780  
 0781  
 0782  
 0783  
 0784  
 0785  
 0786  
 0787  
 0788  
 0789  
 0790  
 0791  
 0792  
 0793  
 0794  
 0795  
 0796  
 0797  
 0798  
 0799  
 0800  
 0801  
 0802  
 0803  
 0804  
 0805  
 0806  
 0807  
 0808  
 0809  
 0810  
 0811  
 0812  
 0813  
 0814  
 0815  
 0816  
 0817  
 0818  
 0819  
 0820  
 0821  
 0822  
 0823  
 0824  
 0825  
 0826  
 0827  
 0828  
 0829  
 0830  
 0831  
 0832  
 0833  
 0834  
 0835  
 0836  
 0837  
 0838  
 0839  
 0840  
 0841  
 0842  
 0843  
 0844  
 0845  
 0846  
 0847  
 0848  
 0849  
 0850  
 0851  
 0852  
 0853  
 0854  
 0855  
 0856  
 0857  
 0858  
 0859  
 0860  
 0861  
 0862  
 0863  
 0864  
 0865  
 0866  
 0867  
 0868  
 0869  
 0870  
 0871  
 0872  
 0873  
 0874  
 0875  
 0876  
 0877  
 0878  
 0879  
 0880  
 0881  
 0882  
 0883  
 0884  
 0885  
 0886  
 0887  
 0888  
 0889  
 0890  
 0891  
 0892  
 0893  
 0894  
 0895  
 0896  
 0897  
 0898  
 0899  
 0900  
 0901  
 0902  
 0903  
 0904  
 0905  
 0906  
 0907  
 0908  
 0909  
 0910  
 0911  
 0912  
 0913  
 0914  
 0915  
 0916  
 0917  
 0918  
 0919  
 0920  
 0921  
 0922  
 0923  
 0924  
 0925  
 0926  
 0927  
 0928  
 0929  
 0930  
 0931  
 0932  
 0933  
 0934  
 0935  
 0936  
 0937  
 0938  
 0939  
 0940  
 0941  
 0942  
 0943  
 0944  
 0945  
 0946  
 0947  
 0948  
 0949  
 0950  
 0951  
 0952  
 0953  
 0954  
 0955  
 0956  
 0957  
 0958  
 0959  
 0960  
 0961  
 0962  
 0963  
 0964  
 0965  
 0966  
 0967  
 0968  
 0969  
 0970  
 0971  
 0972  
 0973  
 0974  
 0975  
 0976  
 0977  
 0978  
 0979  
 0980  
 0981  
 0982  
 0983  
 0984  
 0985  
 0986  
 0987  
 0988  
 0989  
 0990  
 0991  
 0992  
 0993  
 0994  
 0995  
 0996  
 0997  
 0998  
 0999  
 1000

```

1370 DATA 0A,00,FIM
1380 CLS:PRINT "CARREGANDO ROTINA V-BSAV"
E"
1390 FOR I=&HD000 TO &HD109:READ AS:POKE
I,VAL("&H"+AS):NEXT I:PRINT "ROTINA CAR
REGADA." :END

```

Uso: após salvar o seu programa em binário rebobine a fita e comande.

```
DEFUSR=&HD000:USR(0)
```

Naturalmente o seu programa não pode estar entre as posições &HD000 e &HD110 (por quê?) A qualquer instante a verificação pode ser interrompida por CONTROL+STOP.



#### BIBLIOGRAFIA RECOMENDADA.

Programação Avançada em MSX - capítulo 4

## 4.6 - GRAVADOR DE SCREEN 2

Desta vez é um programinha que guarda a tela gráfica (SCREEN 2) em fita cassete, deixando de ser privilégio dos possuidores de disk drives essa operação.

Após carregar a rotina em L M, basta chamar a rotina desejada pela função USA. Naturalmente a função deve ser pré-programada pois o computador apaga a SCREEN 2 no modo de comando

```
1000 REM -----
1010 REM COPIADOR SCREEN 2(--)>FITA
1020 REM JANLEIRO 1988 - THE PILOT
1030 REM --
1040 DATA F3,DB,98,AF,D3,99,D3,99
1050 DATA CD,EA,00,DB,CD,ED,00,CD
1060 DATA ED,00,CD,ED,00,01,00,3B
1070 DATA DB,98,C5,CD,ED,00,C1,D8
1080 DATA 08,78,B1,20,F3,CD,F0,00
1090 DATA C9,21,00,20,01,00,18,3E
1100 DATA 1F,CD,56,00,F3,DB,98,AF
1110 DATA D3,99,D3,99,CD,E1,00,D8
1120 DATA CD,E4,00,CD,E4,00,CD,E4
1130 DATA 00,CD,E4,00,01,00,3B,C5
1140 DATA CD,E4,00,C1,D8,D3,98,0B
1150 DATA 78,B1,20,F3,CD,F7,00,C9
1160 DATA FIM
1170 CLS:FOR I=&HD000 TO &HD05F:READ A$:
POKE I,VAL("&H"+A$):NEXT
1180 PRINT "SCREEN 2( )>FITA":PRINT:PRINT:
I="&H0000":PRINT"LER: &HD029"
&PRINT:END
```

MOUL = 385

## BIBLIOGRAFIA RECOMENDADA.

Programação Avançada em MSX - capítulo 4.



## DICAS PARA A IMPRESSORA

O acesso à impressora nos micros MSX pode ser realizado de várias maneiras diferentes sendo que o usuário pode interferir em algumas delas.

As dicas deste capítulo desvendam aos usuários comuns recursos disponíveis somente através da linguagem de Máquina ou de programas em BASIC bem construídos.

5 1 - Eco na impressora	.....	118
5 2 - Transformando PRINT em LPRINT		119
5 3 - Impressão dupla	...	120
5 4 - HEXA-PRINTER e HEXA-SCREEN	..	122
5 5 - Filtro genérico	, ,	123
5 6 - Impedindo o uso da impressora	..	126
5 7 - Cópia gráfica	.....	127
5 8 - Caracteres de tados	, , .....	130
5 9 - Strings em modo gráfico	, ,	131
5 A - Máquina de escrever	, , . . . . .	132
5 B - Impressor de programas	, ,	133

## 5.1 - ECO NA IMPRESSORA

O MSX possui duas rotinas do BIOS para o envio de dados à impressora. Uma delas, a LPTOUT, pode ser facilmente usada para reproduzir o que for enviado para a tela diretamente na impressora. Uma aplicação desse tipo de recurso pode ser facilmente entendida se pensarmos em programas que apresentam resultados apenas na tela. Para fazê-los enviar os resultados para a impressora teríamos normalmente que alterá-los por inteiro. Obviamente, a maneira mais fácil é usar o programa apresentado a seguir.

```
100 CLEAR 200,&HE000
110 FOR F=&HE000 TO &HE081
120 READ A$: POKE F,VAL("&H"+A$)
130 NEXT F : DLIUSR0=&HE000
140 POKE 0,USR0,0) : END
150 REM
160 REM DADOS
170 REM
1000 DATA FD,21,A4,FD,FD,36,00,C3
1010 DATA FD,36,01,11,FD,36,02,F0
1020 DATA C9,FE,7A,30,31,47,3A,B0
1030 DATA E0,FE,01,78,28,3F,47,3A
1040 DATA B0,E0,FE,02,78,28,19,47
1050 DATA 3A,B0,E0,FE,03,78,28,49
1060 DATA 47,3A,B0,E0,FE,04,78,28
1070 DATA 07,FE,30,38,09,LD,A5,00
1080 DATA 21,B0,F0,36,00,L9,FE,18
1090 DATA 28,01,FE,00,28,1F,FE,0A
1100 DATA 28,EB,3E,20,18,E7,21,80
1110 DATA E0,36,01,18,EB,FE,59,28
1120 DATA 0A,FE,4E,28,0D,FE,4F,28
1130 DATA 09,18,D5,21,B0,E0,36,03
1140 DATA 18,D3,21,B0,E0,36,02,18
1150 DATA CC,21,B0,E0,36,04,18,C5
1160 DATA 00,00,52,45,4E,41,54,4F
```

01F  
02F  
03F  
04F  
05F  
06F  
07F  
08F  
09F  
10F  
11F  
12F  
13F  
14F  
15F  
16F  
17F  
18F  
19F  
20F  
21F  
22F  
23F  
24F  
25F  
26F  
27F  
28F  
29F  
30F  
31F  
32F  
33F  
34F  
35F  
36F  
37F  
38F  
39F  
40F  
41F  
42F  
43F  
44F  
45F  
46F  
47F  
48F  
49F  
50F  
51F  
52F  
53F  
54F  
55F  
56F  
57F  
58F  
59F  
60F  
61F  
62F  
63F  
64F  
65F  
66F  
67F  
68F  
69F  
70F  
71F  
72F  
73F  
74F  
75F  
76F  
77F  
78F  
79F  
80F  
81F  
82F  
83F  
84F  
85F  
86F  
87F  
88F  
89F  
90F  
91F  
92F  
93F  
94F  
95F  
96F  
97F  
98F  
99F

TOTAL = 140H

Após executá-lo, pode-se apagá-lo da memória com o comando NEW. Entretanto é conveniente salvá-lo previamente em disco ou em fita.

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulo 6 e página 159.  
Programação Avançada em MSX - capítulo 3 e página 149

## 5.2 - TRANSFORMANDO PRINT EM LPRINT

O pequeno programa apresentado a seguir deve ser inserido no final de programas maiores. Para executá-lo, deve-se comandar:

RUN 65000

(e RETLNR)

Ao ser executado ele "varre" o programa principal à procura de instruções PRINT e as substitui por instruções LPRINT. Isso pode ser muito útil para redirecionar a saída de dados da tela para a impressora, porém o programa não deve ter instruções PRINT \$. pois serão transformadas em LPRINT \$. Ocasionalmente erros de sintaxe. O programa também não deverá conter números cujo código compactado seja igual a token do PRINT (145, pois nesse caso eles terão seus valores alterados (observe como procedemos para evitar isso na linha 65010 do programa, ao invés de 145 escrevemos 100+45)).

65000	REM	214
65001	REM Muda PRINT p/ LPRINT	253
65002	REM	254
65003	EI = 32769	255
65004	B1 = PEEK(EI) : B2 = PEEK(EI+1)	256
65005	B3 = PEEK(EI+2) : B4 = PEEK(EI+3)	257
65006	PL = B1 + 256*B2	258
65007	NL = B3 + 256*B4	259
65008	IF B1=0 AND B2=0 THEN END	260
65009	FOR F=EI+4 TO PL-2	261
65010	IF PEEK(F)<>(100+45) THEN 65012	262
65011	POKE F,157	263
65012	NEXT F	264
65013	EI=PL	265
65014	GOTO 65004	266

TOTAL = 696

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX capítulo 1  
Programação Avançada em MSX capítulo 2

### 5.3 - IMPRESSÃO DUPLA

O programa apresentado a seguir gera uma rotina em linguagem de Máquina a partir do endereço &HE000 capaz de fazer com que a impressão de listagens e outros dados se, a feita com dupla passagem da cabeça de impressão. Com isso os textos ficarão mais legíveis e com maior contraste.

Após digitar e gravar o programa você pode apagá-lo da memória com o comando NEW.

Caso você deseje desativar a dupla impressão, basta comandar:

POKE &HFFB6,&HC9

Para ativá-la novamente, uma vez que tenha sido desativada, basta comandar:

POKE &HFFB6,&HC3

Note que o programa em BASIC ao ser rodado pede a quantidade de caracteres a serem impressos em cada linha. Uma vez especificado esse parâmetro, sempre que a dupla impressão estiver ativa, as linhas serão impressas com essa largura.

A rotina em Linguagem de Máquina não funcionará como esperado se o número de colunas especificado for maior que o número de colunas da impressora.

```
100 SCREEN 0 : CLEAR 200,&HE000
110 PRINT : PRINT : PRINT
120 PRINT "  Caracteres por linha ";
130 INPUT C
140 IF C<1 OR C>255 THEN RUN
150 FOR F=&HE000 TO &HE06F
160   READ A$:POKE F,VAL("&H"+A$)
170 NEXT F
180 FOR F &HE070 TO &HE173
190   POKE F,0
200 NEXT F
210 POKE &HE02A,C
220 DEFUSR=&HE000 : POKE 0,USR0(0)
230 END
240 DATA F3,21,0E,E0,22,B7,FF,3E
250 DATA C3,32,B6,FF,F8,C9,F5,C5
260 DATA D5,E5,FE,0D,28,52,FE,0A
270 DATA 28,13,E0,4B,6F,E1,21,70
280 DATA E0,09,03,ED,43,6F,E1,77
290 DATA 79,FE,FF,20,3B,3E,C9,32
```

0000  
0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
000A  
000B  
000C  
000D  
000E  
000F  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
001A  
001B  
001C  
001D  
001E  
001F  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
002A  
002B  
002C  
002D  
002E  
002F  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0039  
003A  
003B  
003C  
003D  
003E  
003F  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
004A  
004B  
004C  
004D  
004E  
004F  
0050  
0051  
0052  
0053  
0054  
0055  
0056  
0057  
0058  
0059  
005A  
005B  
005C  
005D  
005E  
005F  
0060  
0061  
0062  
0063  
0064  
0065  
0066  
0067  
0068  
0069  
006A  
006B  
006C  
006D  
006E  
006F  
0070  
0071  
0072  
0073  
0074  
0075  
0076  
0077  
0078  
0079  
007A  
007B  
007C  
007D  
007E  
007F  
0080  
0081  
0082  
0083  
0084  
0085  
0086  
0087  
0088  
0089  
008A  
008B  
008C  
008D  
008E  
008F  
0090  
0091  
0092  
0093  
0094  
0095  
0096  
0097  
0098  
0099  
009A  
009B  
009C  
009D  
009E  
009F  
00A0  
00A1  
00A2  
00A3  
00A4  
00A5  
00A6  
00A7  
00A8  
00A9  
00AA  
00AB  
00AC  
00AD  
00AE  
00AF  
00B0  
00B1  
00B2  
00B3  
00B4  
00B5  
00B6  
00B7  
00B8  
00B9  
00BA  
00BB  
00BC  
00BD  
00BE  
00BF  
00C0  
00C1  
00C2  
00C3  
00C4  
00C5  
00C6  
00C7  
00C8  
00C9  
00CA  
00CB  
00CC  
00CD  
00CE  
00CF  
00D0  
00D1  
00D2  
00D3  
00D4  
00D5  
00D6  
00D7  
00D8  
00D9  
00DA  
00DB  
00DC  
00DD  
00DE  
00DF  
00E0  
00E1  
00E2  
00E3  
00E4  
00E5  
00E6  
00E7  
00E8  
00E9  
00EA  
00EB  
00EC  
00ED  
00EE  
00EF  
00F0  
00F1  
00F2  
00F3  
00F4  
00F5  
00F6  
00F7  
00F8  
00F9  
00FA  
00FB  
00FC  
00FD  
00FE  
00FF

```

300 DATA B6,FF,CD,50,E0,CD,50,E0
310 DATA ED,4B,6F,E1,78,81,3E,0A
320 DATA C4,A5,00,21,00,00,22,6F
330 DATA E1,3E,C3,32,86,FF,18,18
340 DATA ED,4B,6F,E1,78,81,C8,21
350 DATA 70,E0,7E,CD,A5,00,23,00
360 DATA 20,FB,3E,00,CD,A5,00,C9
370 DATA E1,D1,C1,F1,33,33,87,C9

```

```

P28X
P28X
P28X
P28X
P28X
P28X
P28X
P28X

```

```

TOTAL = 0000

```



BIBLIOTECA DE PROGRAMAÇÃO

Programação Avançada em MSX - capítulo 3.



## 5.4 - HEXA-PRINTER E HEXA-SCREEN

Muitas vezes precisamos saber quais caracteres (normalmente de contro e) um dado programa (normalmente em Linguagem de Máquina) está enviando para a impressora. Nesses casos seria muito cômodo se ao invés de imprimir os caracteres propriamente ditos pudessémos imprimir seus códigos hexadecimais. É exatamente isso que o programa listado abaixo faz, com uma opção ainda mais útil para quem não possui impressora, a possibilidade de simular na tela de vídeo a saída de impressora. Digite e grave o programa. Depois rode-o e use algumas vezes o comando LPRINT.

100 FOR F &H0000 TO &H007F	E000
110 READ A\$:POKE F,VAL("&H"+A\$)	E001
120 NEXT F	E002
130 SCREEN 0:LOCATE 0,7,0:KEY OFF	E003
140 PRINT SPC(10);"C 1 ] HEXAPRINTER"	E004
150 PRINT SPC(10);"C 2 ] HEXASCREEN"	E005
160 A\$=INKEY\$	E006
170 IF A\$="1" THEN 260	E007
180 IF A\$="2" THEN 160	E008
190 POKE &H003B,&HA2	E009
200 POKE &H004A,&HA2	E010
210 POKE &H004F,&HA2	E011
220 POKE &H0058,&HC	E012
230 POKE &H005C,&HA2	E013
240 POKE &H0061,&HA2	E014
250 POKE &H0070,&HC	E015
260 DIMUSR &H0000 : S=USR(0)	E016
270 SCREEN 0 : NEW	E017
280 DATA F3,0D,21,B6,FF,0D,36,00	E018
290 DATA C3,0D,36,01,13,03,36,02	E019
300 DATA D0,F8,C9,E5,D5,C5,F5,FE	E020
310 DATA 0A,20,05,21,70,00,36,01	E021
320 DATA 21,06,FF,36,C9,47,3E,F0	E022
330 DATA A0,CB,3F,CB,3F,CB,3F,CB	E023
340 DATA 3F,C6,30,FE,3A,38,02,C6	E024
350 DATA 07,C5,CD,A5,00,C1,3E,0F	E025
360 DATA A0,C6,30,FE,3A,38,02,C6	E026
370 DATA 07,CD,A5,00,3E,20,CD,A5	E027
380 DATA 00,21,70,D0,35,20,0C,36	E028
390 DATA 10,3E,0D,CD,A5,00,3E,0A	E029
400 DATA CD,A5,00,21,B6,FF,36,C3	E030
410 DATA F1,C1,D1,E1,33,33,B7,C9	E031
420 DATA 10,52,52,45,4E,41,54,4F	E032

## 5.5 - FILTRO GENÉRICO

O programa apresentado a seguir permite compatibilizar os caracteres acentuados de seu MSX (Expert 1.1 ou Hobbit) com a sua impressora desde que ela os possua.

Digite e grave o programa a seguir e depois execute-o comandando:

RUN

Ligue a impressora e pressione a tecla RETJRN

Serão impressos os caracteres correspondentes aos códigos de 128 (&H80) a 255 (&HFF) e seus respectivos códigos em hexadecimal. Não se assuste se durante essa impressão sua impressora "agitar" de forma estranha pois pode ser que alguns dos caracteres enviados para ela correspondam a alguns de seus controles como avanço de linha, beep, etc.

Quando a impressão terminar veja o resultado e pressione RETJRN novamente. Serão listadas na tela as linhas de 620 a 710 do programa para que você as altere conforme o resultado obtido na listagem da impressora.

Procure cada caractere das linhas DATA na lista impressa e substitua o código 20 (da atagem original) pelo código que foi impresso.

Faça todas as alterações (não se esqueça de pressionar RETURN após cada uma delas!) comande:

GO TO 360

Com isso será gravado um arquivo de nome "FILTRO.BIN". Ele é seu programa "filtro" e para rodá-lo, basta comandar:

BLOAD "FILTRO.BIN",R

Utilize-o toda vez que você pretender imprimir um texto com acentuação

```
100 POKE &HF417,1:SCREEN 0
110 WIDTH 30:KEYOFF
120 LOCATE 0,10:PRINT "  PREPARE SUA";
130 PRINT " IMPRESSORA E QUANDO ES-";
140 PRINT " TIVER PRONTO APERTE A TECLA";
150 PRINT " (RETURN).";
160 AS=INPUT$(1)
170 IF ASC(AS)<>13 THEN 160
```

005  
010  
015  
020  
025  
030  
035  
040  
045  
050  
055  
060  
065  
070  
075  
080  
085  
090  
095  
100  
105  
110  
115  
120  
125  
130  
135  
140  
145  
150  
155  
160  
165  
170  
175  
180  
185  
190  
195  
200  
205  
210  
215  
220  
225  
230  
235  
240  
245  
250  
255  
260  
265  
270  
275  
280  
285  
290  
295  
300  
305  
310  
315  
320  
325  
330  
335  
340  
345  
350  
355  
360  
365  
370  
375  
380  
385  
390  
395  
400  
405  
410  
415  
420  
425  
430  
435  
440  
445  
450  
455  
460  
465  
470  
475  
480  
485  
490  
495  
500  
505  
510  
515  
520  
525  
530  
535  
540  
545  
550  
555  
560  
565  
570  
575  
580  
585  
590  
595  
600  
605  
610  
615  
620  
625  
630  
635  
640  
645  
650  
655  
660  
665  
670  
675  
680  
685  
690  
695  
700  
705  
710  
715  
720  
725  
730  
735  
740  
745  
750  
755  
760  
765  
770  
775  
780  
785  
790  
795  
800  
805  
810  
815  
820  
825  
830  
835  
840  
845  
850  
855  
860  
865  
870  
875  
880  
885  
890  
895  
900  
905  
910  
915  
920  
925  
930  
935  
940  
945  
950  
955  
960  
965  
970  
975  
980  
985  
990  
995  
1000

```

180 FOR F= 128 TO 255
190 LPRINT CHR$(F);" = ";HEX$(F)
200 NEXT
210 CLS
220 LOCATE 0,7:PRINT"  VEJA O QUE SAIU";
230 PRINT" NA IMPRESSORA E ";
240 PRINT" ALTERE AS LINHAS DATA."
250 PRINT:PRINT" DEPOIS DE ALTERADAS";
260 PRINT", COMANDE ";
270 PRINT"GOTO 300":PRINT
280 PRINT" APERTE A TECLA <RETURN>";
290 PRINT" PARA COMEÇAR"
300 AS=INPUT$(1)
310 IF ASC(AS)<>13 THEN 300
320 CLS:PRINT "  NÃO SE ESQUEÇA DO";
330 PRINT " <RETURN> APÓS AS  ALTER";
340 PRINT "AÇÕES E DE COMANDAR GOTO ";
350 PRINT "360.":PRINT:LIST 620 710:END
360 CLS
370 LOCATE 0,10:PRINT"PREPARE O DISCO";
380 PRINT" PARA GRAVAR O PROGRAMA"
390 PRINT"FILTRO E TECLA RETURN ";
400 PRINT" QUANDO PRONTO"
410 AS=INPUT$(1)
420 IF ASC(AS)<>13 THEN 410
430 RESTORE 620
440 FOR L=0 TO 55
450 READ Z$
460 Z$ "&H"+RIGHT$(Z$,2)
470 POKE &HD024+L,VAL(Z$)
480 NEXT
490 RESTORE 720
500 FOR E &HD000 TO &HD025
510 READ Z$
520 POKE E,VAL("&H"+Z$)
530 NEXT
540 SAVE "FILTRO.BIN",&HD000,&HD05D
550 CLS
560 LOCATE 0,8:PRINT"  O PROGRAMA";
570 PRINT" ESTÁ GRAVADO.":PRINT
580 PRINT" PARA RODAR LO, COMANDE:"
590 PRINT:PRINT" BLOAD";CHR$(34);
600 PRINT"FILTRO.BIN";CHR$(34);",R"
610 PRINT:PRINT:END
620 DATA C=20 u=20, e=20, a=20 A=20, a=20
630 DATA =20 c=20, e=20, f=20 d=20, o=20
640 DATA A=20 e=20, d=20 A=20, e=20 x=20
650 DATA f=20, d=20, o=20 d=20 i=20 u=20
660 DATA y=20 o=20, u=20, t=20, e=20, x=20

```

```

0000
0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999

```

1000  
101  
10123  
10135  
10138  
10139  
10144  
10145  
10146  
10147  
10148

Como exemplo, apresentamos a seguir as planilhas DATA de 620 a 710 preenchidas para compatibilizar um Expert 11 com uma impressora Mônica El6030.



## 5.6 - IMPEDINDO O USO DA IMPRESSORA

Muitos programadores tentam proteger ao menos a originalidade de seus programas BASIC dos milhares de piratas amadores que farteiam por este país simplesmente desativando a listagem do mesmo na tela. Lado engano o daqueles que pensam que isso é eficaz! Basta comandar LLIST e a listagem será enviada para a impressora. Se, entretanto, o programa ao ser carregado, desativar o uso da impressora o problema estará resolvido. Para isso, basta inserir o código &HC3 na posição de memória &HFFB6. Experimente rodar o programinha exemplo listado abaixo e depois tente enviá-lo para a impressora.

```
10 POKE &HFFB6,&HC3  
20 REM Tente me listar numa impressora  
30 PRINT "Comande LLIST!"  
40 END
```

1000000 = 1000000

## BIBLIOGRAFIA RECOMENDADA:

Programação Avançada em MSX - página 148.

### 5.7 - Cópia Gráfica

O programa listado a seguir imprime uma cópia fiel da tela presente no vídeo do micro mediante o simples pressionamento da tecla ESC.

5 Digite-o e grave-o. Depois execute-o. Com isso a  
6 rotina em Linguagem de Máquina estará pronta para ser  
7 usada na memória do micro.

```

100 FOR F=&HE000 TO &HE25F
110 READ AS : POKE F,VAL("&H"+AS)
120 NEXT F
130 DEFUSR=&HE000 : S=USR(0)
140 SCREEN 0 : NEW
150 DATA 3A,CC,FD,FE,C9,C0,21,12
160 DATA E0,22,CD,FD,3E,CD,32,CC
170 DATA FD,C9,FE,3A,C0,F5,C5,D5
180 DATA E5,ED,73,59,E2,0E,00,3A
190 DATA AF,FC,B7,21,F0,00,11,2B
200 DATA 06,2B,06,21,00,01,11,20
210 DATA 0B,3E,1B,CD,97,E0,3E,4B
220 DATA CD,97,E0,7D,CD,97,E0,7C
230 DATA CD,97,E0,06,00,CD,A1,E0
240 DATA D5,C5,21,5B,E2,42,11,0B
250 DATA 00,C5,E5,06,0B,7E,FE,0B
260 DATA 3F,CB,11,19,10,F7,79,CD
270 DATA 97,F0,E1,C1,23,10,EA,C1
280 DATA D1,04,7B,BB,20,D7,3E,0D
290 DATA CD,97,E0,3E,1B,CD,97,E0
300 DATA 3E,41,CD,97,E0,3E,0B,CD
310 DATA 97,E0,3E,0A,CD,97,E0,3E
320 DATA 1B,CD,97,E0,0C,79,FE,1B
330 DATA 20,8D,E1,D1,C1,F1,C9,CD
340 DATA A5,00,D0,ED,7B,59,E2,1B
350 DATA F1,C5,D5,E5,FD,E5,21,5B
360 DATA E2,3E,40,36,00,23,10,20
370 DATA FA,3A,AF,FC,B7,F5,C5,C4
380 DATA 73,E1,C1,69,26,00,29,29
390 DATA 29,5D,54,29,29,F1,F5,20
400 DATA 01,19,5B,19,EB,D6,02,79
410 DATA 01,00,00,2A,24,F9,E5,2A
420 DATA 22,F9,3B,19,20,0C,2A,CB
430 DATA F3,E3,2A,C7,F3,E6,1B,47
440 DATA 1B,0B,2A,D5,F3,E3,2A,D1
450 DATA F3,07,E6,06,4F,19,CD,4A
460 DATA 00,6F,26,00,29,29,29,09
470 DATA E8,FD,E1,FD,19,2A,C9,F3
480 DATA 19,0F,0F,0F,E6,1F,4F,06
490 DATA 00,3A,E6,F3,57,E6,0F,5F

```

[illegible]



```

30 CIRCLE (128,80),80,15,,,80/F
40 CIRCLE (128,80),80,15,,,F/80
50 NEXT F
60 LINE (128,160)-(128,0)
70 LINE (48,80)-(208,80)
80 GOTO 80

```

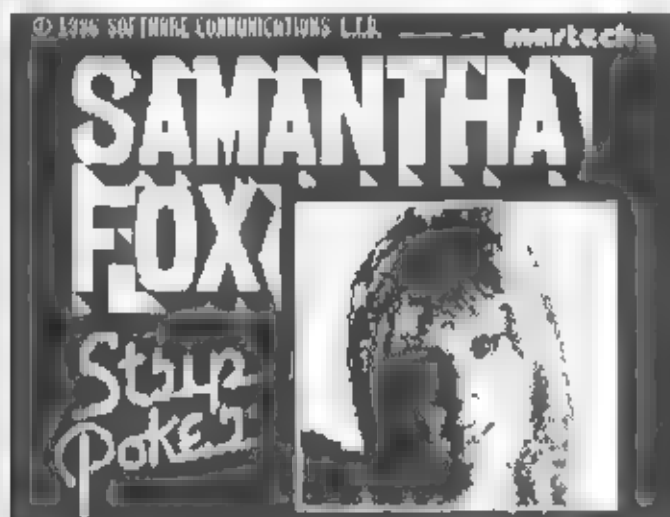
```

001
13A5
15AC
1F18
263D
2905

```

TOTAL = 2945

O programa copia para a impressora todas as regões da tela (qualquer uma das SCREEN's) cuja cor possua um código superior a 7 inclusive as SPRITES



# BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX - capítulo 6.  
 Programação Avançada em MSX - capítulo 3.



## 5.8 - CARACTERES DEITADOS

O programa listado adiante gera uma rotina em L.M. que imprime uma string em modo gráfico, porém gira previamente, seus caracteres de 90°.

Isso pode ser de grande utilidade para produzir mensagens que devem ser dadas "em pé", de cima para baixo.

A string a ser impressa deve ser passada como parâmetro na função USR.

10	'*****'	0000
20	'* BY THE DOCTOR LUZ *	0001
30	'*****'	0002
40	FOR I=&HC000 TO &HC0BA	0003
50	READ A\$:POKE L,VAL("&H"+A\$)	0004
60	NEXT	0005
70	DEF USR=&HC000	0006
80	PRINT"RODAR PROGRAMA ?"	0007
90	A\$=INPUT\$(1)	0008
100	IF A\$="S" THEN GOTO 200 ELSE END	0009
200	PRINT"DIGITE A STRING A SER IMPRESSA"	0010
210	INPUT A\$	0011
220	A\$=USR(A\$)	0012
230	GOTO 80	0013
1000	DATA 2A,F8,F7,46,05,C0,3F,C0	0014
1010	DATA 73,7E,5F,23,7E,57,F8,E5	0015
1020	DATA E1,23,E5,7E,C5,21,8F,18	0016
1030	DATA 5F,16,00,C8,23,C8,12,C8	0017
1040	DATA 23,C8,12,C8,23,C8,12,AF	0018
1050	DATA ED,5A,EB,C0,61,C0,06,08	0019
1060	DATA 21,00,C2,7E,C0,A5,00,23	0020
1070	DATA 10,F9,C1,10,D3,E1,C9,3E	0021
1080	DATA 1B,C0,A5,00,3E,4B,C0,A5	0022
1090	DATA 00,58,16,00,C8,23,C8,12	0023
1100	DATA C8,23,C8,12,C8,23,C8,12	0024
1110	DATA 78,C0,A5,00,7A,C0,A5,00	0025
1120	DATA C9,06,08,C5,21,00,C2,06	0026
1130	DATA 08,1A,17,C8,16,23,10,FA	0027
1140	DATA C1,13,10,EF,C9,C0,ED,00	0028
1150	DATA D1,E1,3B,07,23,E7,20,F2	0029
1160	DATA C3,F0,00,21,30,71,C3,1A	0030
1170	DATA 70,7D,C5	0031

## BIBLIOGRAFIA RECOMENDADA.

Linguagem de Máquina MSX - páginas 140 a 143.

## 5.9 - STRINGS EM MODO GRÁFICO

O programa a seguir é quase igual ao da dica 5.8, porém a impressão da string é feita sem que os caracteres seja girados.

Lembre-se que tanto esta dica quanto a anterior só funcionarão se a impressora conectada ao micro entrar em modo gráfico segundo o padrão EPSON e tiver 9 agulhas em sua cabeça de impressão.

```
10 *****
20 * BY THE DOCTOR LUZ *
30 *****
40 FOR L=&HC000 TO &HC072
50 READ A$:POKE L,VAL("&H"+A$)
60 NEXT
70 DEF USR=&HC000
80 PRINT"RODAR PROGRAMA ?"
90 A$=INPUT$(1)
100 IF A$="S" THEN GOTO 200 ELSE END
200 PRINT"DIGITE A STRING A SER IMPRESSA"
210 INPUT A$
220 A$=USR(A$)
230 GOTO 80
5000 DATA 2A,F8,F7,46,05,CD,3C,C0
5010 DATA 23,7E,5F,23,7E,57,6B,26
5020 DATA 00,ED,5A,E5,E1,2B,E5,7E
5030 DATA C5,21,C6,1B,5F,16,00,CB
5040 DATA 23,CB,12,CB,23,CB,12,CB
5050 DATA 23,CB,12,AF,ED,5A,06,0B
5060 DATA 7E,C0,A5,00,2B,10,F9,C1
5070 DATA 10,DA,E1,C9,3E,1B,C0,A5
5080 DATA 00,3E,4B,C0,A5,00,5B,16
5090 DATA 00,CB,23,CB,12,CB,23,CB
5100 DATA 12,CB,23,CB,12,7B,C0,A5
5110 DATA 00,7A,C0,A5,00,C9,06,0B
5120 DATA C5,21,00,C2,06,0B,1A,17
5130 DATA CB,16,23,10,FA,C1,13,10
5140 DATA EF,C9,10
```

TOTAL = 1009

### BIBLIOGRAFIA RECOMENDADA:

Linguagem de Máquina MSX - página 143

## 5.A - MÁQUINA DE ESCRIVER

O pequeno programa listado a seguir, após ser executado, faz com que todos os caracteres digitados no teclado apareçam na tela e também na impressora. Deste modo pode-se usar a impressora quase como uma máquina de escrever.

```
100 KEY1,CHR$(13)+CHR$(10)
110 SCREEN 0 : WIDTH 40
120 PRINT "F1 avança 1 linha!"
130 PRINT "HOME/CLS avança uma página!"
140 FOR F=&HE000 TO &HE00E
150   READ AS=POKE F,VAL("&H"+AS)
160 NEXT F : DEFUSR0=&HF000
170 POKE 0,USR0(0) : END
180 DATA CD,9F,00,CD,A2,00,CD,4D
190 DATA 01,CD,07,00,30,F2,C9,00
```

801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000

1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2820 2821 2822 2823 2824 2825 2826 2827 2828 2829 2830 2831 2832 2833 2834 2835 2836 2837 2838 2839 2840 2841 2842 2843 2844 2845 2846 2847 2848 2849 2850 2851 2852 2853 2854 2855 2856 2857 2858 2859 2860 2861 2862 2863 2864 2865 2866 2867 2868 2869 2870 2871 2872 2873 2874 2875 2876 2877 2878 2879 2880 2881 2882 2883 2884 2885 2886 2887 2888 2889 2890 2891 2892 2893 2894 2895 2896 2897 2898 2899 2900 2901 2902 2903 2904 2905 2906 2907 2908 2909 2910 2911 2912 2913 2914 2915 2916 2917 2918 2919 2920 2921 2922 2923 2924 2925 2926 2927 2928 2929 2930 2931 2932 2933 2934 2935 2936 2937 2938 2939 2940 2941 2942 2943 2944 2945 2946 2947 2948 2949 2950 2951 2952 2953 2954 2955 2956 2957 2958 2959 2960 2961 2962 2963 2964 2965 2966 2967 2968 2969 2970 2971 2972 2973 2974 2975 2976 2977 2978 2979 2980 2981 2982 2983 2984 2985 2986 2987 2988 2989 2990 2991 2992 2993 2994 2995 2996 2997 2998 2999 3000 3001 3002 3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018 3019 3020 3021 3022 3023 3024 3025 3026 3027 3028 3029 3030 3031 3032 3033 3034 3035 3036 3037 3038 3039 3040 3041 3042 3043 3044 3045 3046 3047 3048 3049 3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3067 3068 3069 3070 3071 3072 3073 3074 3075 3076 3077 3078 3079 3080 3081 3082 3083 3084 3085 3086 3087 3088 3089 3090 3091 3092 3093 3094 3095 3096 3097 3098 3099 3100 3101 3102 3103 3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114 3115 3116 3117 3118 3119 3120 3121 3122 3123 3124 3125 3126 3127 3128 3129 3130 3131 3132 3133 3134 3135 3136 3137 3138 3139 3140 3141 3142 3143 3144 3145 3146 3147 3148 3149 3150 3151 3152 3153 3154 3155 3156 3157 3158 3159 3160 3161 3162 3163 3164 3165 3166 3167 3168 3169 3170 3171 3172 3173 3174 3175 3176 3177 3178 3179 3180 3181 3182 3183 3184 3185 3186 3187 3188 3189 3190 3191 3192 3193 3194 3195 3196 3197 3198 3

## 5.2 - IMPRESSOR DE PROGRAMAS

O programa apresentado a seguir permite a impressão de listagens de programas sem que eles passem pelo "picote" do formulário contínuo. Ele só funcionará se o formulário for o de uso padrão com 86 linhas de impressão, e se o entrelinhamento da impressora não tiver sido alterado.

Para usar o programa, digite-o e rode-o. Ele irá gerar e gravar o programa em Linguagem de Máquina responsável pelo "salto" sobre o "picote". Antes de comandar LLIST para listar um programa de mais de 66 linhas, carregue a rotina em Linguagem de Máquina comandando:

```
LOAD"IMPRE.BIN",R
```

Depois pode-se usar o LLIST normalmente.

```
100 FOR FX=&HE000 TO &HE03A
110   READ A$:POKE FX,VAL("&H"+A$)
120 NEXT FX:DEFUSR0=&HE000
130 POKE 0,USR0(0)
140 BSAVE"IMPRE.BIN",&HE000,&HE03A
150 END
160 DATA F3,3E,C3,32,B6,FF,21,0E
170 DATA E0,22,87,FF,FB,C9,FE,0A
180 DATA C0,3A,3A,E0,3C,32,3A,E0
190 DATA FE,3F,3E,0A,C0,3E,C9,32
200 DATA B6,FF,3E,0A,CD,AS,00,CD
210 DATA AS,00,CD,AS,00,CD,AS,00
220 DATA 3E,C3,32,B6,FF,AF,32,3A
230 DATA E0,C9,00,00,00,00,00,00
```

1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2260  
2261  
2262  
2263  
2264  
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278  
2279  
2280  
2281  
2282  
2283  
2284  
2285  
2286  
2287  
2288  
2289  
2290  
2291  
2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312  
2313  
2314  
2315  
2316  
2317  
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338  
2339  
2340  
2341  
2342  
2343  
2344  
2345  
2346  
2347  
2348  
2349  
2350  
2351  
2352  
2353  
2354  
2355  
2356  
2357  
2358  
2359  
2360  
2361  
2362  
2363  
2364  
2365  
2366  
2367  
2368  
2369  
2370  
2371  
2372  
2373  
2374  
2375  
2376  
2377  
2378  
2379  
2380  
2381  
2382  
2383  
2384  
2385  
2386  
2387  
2388  
2389  
2390  
2391  
2392  
2393  
2394  
2395  
2396  
2397  
2398  
2399  
2400  
2401  
2402  
2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419  
2420  
2421  
2422  
2423  
2424  
2425  
2426  
2427  
2428  
2429  
2430  
2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445  
2446  
2447  
2448  
2449  
2450  
2451  
2452  
2453  
2454  
2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464  
2465  
2466  
2467  
2468  
2469  
2470  
2471  
2472  
2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480  
2481  
2482  
2483  
2484  
2485  
2486  
2487  
2488  
2489  
2490  
2491  
2492  
2493  
2494  
2495  
2496  
2497  
2498  
2499  
2500  
2501  
2502  
2503  
2504  
2505  
2506  
2507  
2508  
2509  
2510  
2511  
2512  
2513  
2514  
2515  
2516  
2517  
2518  
2519  
2520  
2521  
2522  
2523  
2524  
2525  
2526  
2527  
2528  
2529  
2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537  
2538  
2539  
2540  
2541  
2542  
2543  
2544  
2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591  
2592  
2593  
2594  
2595  
2596  
2597  
2598  
2599  
2600  
2601  
2602  
2603  
2604  
2605  
2606  
2607  
2608  
2609  
2610  
2611  
2612  
2613  
2614  
2615  
2616  
2617  
2618  
2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672  
2673  
2674  
2675  
2676  
2677  
2678  
2679  
2680  
2681  
2682  
2683  
2684  
2685  
2686  
2687  
2688  
2689  
2690  
2691  
2692  
2693  
2694  
2695  
2696  
2697  
2698  
2699  
2700  
2701  
2702  
2703  
2704  
2705  
2706  
2707  
2708  
2709  
2710  
2711  
2712  
2713  
2714  
2715  
2716  
2717  
2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729  
2730  
2731  
2732  
2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753  
2754  
2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769  
2770  
2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789  
2790  
2791  
2792  
2793  
2794  
2795  
2796  
2797  
2798  
2799  
2800  
2801  
2802  
2803  
2804  
2805  
2806  
2807  
2808  
2809  
2810  
2811  
2812  
2813  
2814  
2815  
2816  
2817  
2818  
2819  
2820  
2821  
2822  
2823  
2824  
2825  
2826  
2827  
2828  
2829  
2830  
2831  
2832  
2833  
2834  
2835  
2836  
2837  
2838  
2839  
2840  
2841  
2842  
2843  
2844  
2845  
2846  
2847  
2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857  
2858  
2859  
2860  
2861  
2862  
2863  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891  
2892  
2893  
2894  
2895  
2896  
2897  
2898  
2899  
2900  
2901  
2902  
2903  
2904  
2905  
2906  
2907  
2908  
2909  
2910  
2911  
2912  
2913  
2914  
2915  
2916  
2917  
2918  
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935  
2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950



## DICAS PARA O DRIVE

O disk drive é sem dúvida um dos mais importantes periféricos no sistema MSX. Dentre as máquinas de 8 bits provavelmente a que consegue um melhor aproveitamento dos drives são as do padrão MSX. Vamos ver alguns recursos do drive a partir do BASIC e do MSXDOS.

Aconselhamos que a numeração dos programas seja seguida à risca, pois em alguns casos pode-se tirar proveito disso. A sequência de programas das dicas entre a B.2 e a B.6 pode ser emendada através de comandos "MERGE" constituindo-se num único programa utilitário de disco. Leia mais sobre isso nas explicações da dica B.2.

A bibliografia recomendada para todas estas dicas é a mesma:

Usando o Disk Drive no MSX,  
Sistema de Disco para MSX e  
Drives Leopard de 5 1/2"

B 1 - Personalizando disquetes	... 135
B 2 - Cabeçalhos de arquivos	... 136
B 3 - Características de discos	... 139
B 4 - Mapa de discos	... 141
B 5 - Leitor de setores	... 145
B 6 - Leitor de arquivos	... 147
B 7 - Programador de funções	... 149
B 8 - CLS em MSXDOS	... 151
B 9 - BEEP em MSXDOS	... 152
B A - "SWAP" de arquivos em MSXDOS	... 152

## 6.1 - PERSONALIZANDO DISQUETES

O pequeno programa apresentado a seguir grava uma mensagem de até 512 bytes no último setor de um disco.

Você pode usá-lo para "marcar" seus discos com seu nome ou com qualquer outra mensagem.

O texto da mensagem deve ser inserido nas linhas DATA's ao fim do programa e deve terminar sempre com o caractere "0".

100	XS=DISKIS(0,0)	0002
110	EN=PEEK(&HF351)+256*PEEK(&HF352)	0006
120	LS=PEEK(EN+19)+256*PEEK(EN+20)-1	000A
130	READ XS	000E
140	IF XS="0" THEN 190	0012
150	FOR F=EN TO EN+LEN(XS)-1	0016
160	POKE F,ASC(MID\$(XS,F-EN+1,1))	001A
170	NEXT F : EN=F	001E
180	GOTO 130	0022
190	DISKIS 0,LS	0026
200	END	002A
210	REM	002E
220	REM        Texto a ser gravado	0032
230	REM	0036
240	DATA Este disquete é de	003A
250	DATA propriedade exclusiva	003E
260	DATA de seu próprio dono !	0043
270	DATA 0	0047

## BIBLIOGRAFIA RECOMENDADA.

Usando o Disk Drive no MSX - páginas 115 e 116  
Sistema de Disco para MSX - páginas 64 e 65  
Drives Leopard de 3 " - páginas 98 e 99

## 6.2 - CABEÇALHOS DE ARQUIVOS

O programa apresentado a seguir lê o cabeçalho de programas gravados em disco no formato binário. Após digitá-lo e conferi-lo grave-o no formato ASCII.

2000 REM	102
2010 REM     cabeçalho de arquivos	010
2020 REM     -----	F40
2030 PLAY "S0M500001CH"	110
2040 SCREEN 0 : WIDTH 38 : KEY OFF	110
2050 COLOR 15,1,1 : PRINT : PRINT	2070
2060 PRINT\$PC(9);"CABECALHO DE ARQUIVOS"	0010
2070 PRINT\$PC(9);"-----"	0000
2080 PRINT:FILES:PRINT:PRINT:X\$=""	0000
2090 PRINT"NOME DO ARQUIVO ";:INPUT X\$	0000
2100 IF X\$="" THEN 1000	0000
2110 OPEN X\$ AS #1 LEN=1	0000
2120 FIELD #1,1 AS A\$	0000
2130 IF EOF(1)>0 THEN 2140	0000
2131     CLOSE #1 : KILL X\$ : GOTO 2000	0000
2140 GET #1,1 : IF ASC(A\$) > 54 THEN 2160	0000
2150 PRINT"NÃO É BINARIO":GOTO 2260	0000
2160 RESTORE 2250 : PRINT	0000
2170 FOR F= 1 TO 6 STEP 2	0000
2180 READ B\$ : PRINT "  ";B\$;":";	0000
2190 GET #1,F+2	0000
2200 PRINT "&H";	0000
2210 PRINT RIGHT\$("0"+HEX\$(ASC(A\$)),2);	0000
2220 GET #1,F+1	0000
2230 PRINT RIGHT\$("0"+HEX\$(ASC(A\$)),2)	0000
2240 NEXT F : CLOSE #1	0000
2250 DATA INICIO ,FIM ,EXECUÇÃO	0000

MUTAL = F40

Para usar o programa acima e os programas das dicas de 6.3 a 6.6 será necessário digitar também o programa MENU listado a seguir. Digite-o e grave-o em ASCII com o nome MENU

240 FOR F=1 TO 10 : KEY F,"" : NEXT F	0000
1000 REM	0000
1010 REM MENU PRINCIPAL	0000
1020 REM     -----	0000
1030 PLAY "S0M400005GH"	0000
1040 COLOR 15,1 : SCREEN 0 : WIDTH 40	0000
1050 KEY OFF : CLOSE	0000
1060 FOR F=1 TO 8: KEY (F) ON : NEXT F	0000

1070 LOCATE 9,2,0	1070
1080 PRINT ">>> MENU PRINCIPAL <<<"	1080
1090 PRINT SPC(9);STRING\$(22," ")	1090
1100 PRINT	1100
1110 PRINT SPC(7);" [ F1 ] MENU PRINCIPA	1110
L": PRINT	
1120 PRINT SPC(7);" [ F2 ] CABEÇALHO DE	1120
ARQUIVOS": PRINT	
1130 PRINT SPC(7);" [ F3 ] CARACTERÍSTIC	1130
AS": PRINT	
1140 PRINT SPC(7);" [ F4 ] MAPA DO DISQU	1140
ETE": PRINT	
1150 PRINT SPC(7);" [ F5 ] DUMP DE SETOR	1150
ES": PRINT	
1160 PRINT SPC(7);" [ F6 ] DUMP DE ARQUI	1160
VOS": PRINT	
1170 PRINT SPC(7);" [ F7 ] EDITAR DISCO	1170
": PRINT	
1180 PRINT SPC(7);" [ F8 ] TERMINAR": PR	1180
INT	
1190 ON KEY GOSUB 1000,2000,3000,4000,50	1190
00,6000,7000,8000	
1200 GOTO 1200	1200
2260 PRINT : LOCATE 8,23,0	2260
2270 PRINT">>> TECLA ESPAÇO <<<"	2270
2280 IF STRIG(0) = 0 THEN 2280 ELSE RETURN	2280
1000	
3460 PRINT">>> TECLA ESPAÇO <<<"	3460
3470 IF STRIG(0) = 0 THEN 3470 ELSE RETURN	3470
1000	
4920 LOCATE 8,23,0	4920
4930 PRINT ">>> TECLA ESPAÇO <<<"	4930
4940 IF STRIG(0) = 0 THEN 4940 ELSE RETURN	4940
1000	
5500 LOCATE 8,22,0	5500
5510 PRINT">>> TECLA ESPAÇO <<<"	5510
5520 IF STRIG(0) = 0 THEN 5520 ELSE GOTO 1	5520
000	
6430 LOCATE 8,22,0	6430
6440 PRINT ">>> TECLA ESPAÇO <<<"	6440
6450 IF STRIG(0) = 0 THEN 6450 ELSE GOTO 1	6450
000	
7000 REM	7000
7010 REM Editor d sco	7010
7020 RLM ----	7020
7030 SCREEN 0	7030
7040 PRINT : PRINT : PRINT SPC(10);	7040
7050 PRINT "NÃO IMPLEMENTADA !!!"	7050
7060 PRINT : PRINT : PRINT SPC(10);	7060



7070 PRINT " DIGITE RETURN !!!"	10220
7080 COLOR (F) MOD 15 : F=F+1	10240
7090 IF INKEY\$=CHR\$(13) THEN RUN	10260
7100 BEEP : GOTO 7080	10280
8000 REM	10300
8010 REM     terminar	10320
8020 REM -----	10340
8030 SCREEN 0 : WIDTH 40 : KEY ON	10360
8040 DEFUSR0=&H3E : X=USR0(0)	10380
8050 FOR F=1 TO 8 : KEY(F) OFF : NEXT F	10400
8060 END	10420

Quando você já tiver todos os programas (6.2, 6.3, 6.4, 6.5 e 6.6) digitados e gravados digite a seguinte sequência de comandos:

```

LOAD "MENU"
MERGE "programa 6.2"
MERGE "programa 6.3"
MERGE "programa 6.4"
MERGE "programa 6.5"
MERGE "programa 6.6"

```

Depois grave o programa presente na memória do micro com o nome "D SCOLTI BAS". Rode-o para ver como as dicas operam em conjunto.

A "SOMA TOTAL" do programa completo deverá ser:



## 6.3 - CARACTERÍSTICAS DE DISCOS

A rotina apresentada adiante lê o setor 0 de um disco e fornece as características de formatação do mesmo. Após digitar e conferir a listagem grave-a no formato ASCII para posterior utilização com o comando MERGE.

3000 REM	14E
3010 REM Características do disco	14F
3020 REM -----	
3030 P AY "S0M500002DH"	1884
3040 SCREEN 0 : WIDTH 38 : KEY OFF	2075
3050 PRINT "CARACTERÍSTICAS DO DISQUETE"	2062
3060 PRINT "SIRING\$(27,"-") : PRINT	3000
3070 PRINT "Fornecedor O&M ..... :"	3340
3080 PRINT "Bytes por setor .... :"	3374
3090 PRINT "Setores por bloco .. :"	1736
3100 PRINT "Setores reservados . :"	3143
3110 PRINT "No de F.A.T.'s ..... :"	3144
3120 PRINT "Entr. no diretório . :"	3145
3130 PRINT "Setores no disco ... :"	
3140 PRINT "Tipo de disco ..... :"	3000
3150 PRINT "Setores por F.A.T. .... :"	3313
3160 PRINT "Setores por trilha . :"	0133
3170 PRINT "Faces ..... :"	00FE
3180 PRINT "Setores ocultos .... :"	1343
3190 ATNCKIS(0,0) : C 256	1000
3200 F (X)PEEK(&H352)+PEEK(&H371)	
3210 DIFINA(X) L*PEEK(E+X)+PEEK(E+X 1)	1141
3220 DEFNB(X) PEEK(E+X) : LOCATE 23,3	2198
3230 FOR I=E+3 TO E+10	286
3240 PRINT CHR\$(PEEK(I))	311E
3250 NEXT I	3487
3260 LOCATE 21,4 : PRINT FNA(12)	3050
3270 LOCATE 22,5 : PRINT FNB(13)	4873
3280 LOCATE 22,6 : PRINT FNA(15)	3243
3290 LOCATE 22,7 : PRINT FNB(16)	50EB
3300 LOCATE 22,8 : PRINT FNA(18)	681E
3310 LOCATE 22,9 : PRINT FNA(20)	
3320 LOCATE 23,10: PRINT HEX\$(FNB(21))	882E
3330 LOCATE 22,11: PRINT INA(23)	97CE
3340 LOCATE 22,12: PRINT INA(25)	9720
3350 LOCATE 22,13: PRINT INA(27)	9860
3360 LOCATE 22,14: PRINT INA(29)	00E3
3370 PRINT , "Capacidade total ... :"	3333
3380 BS = FNA(12) : SE = FNA(20)	3333
3390 LOCATE 22,16,0	3333
3400 PRINT USING "#####";SE*BS;	055C

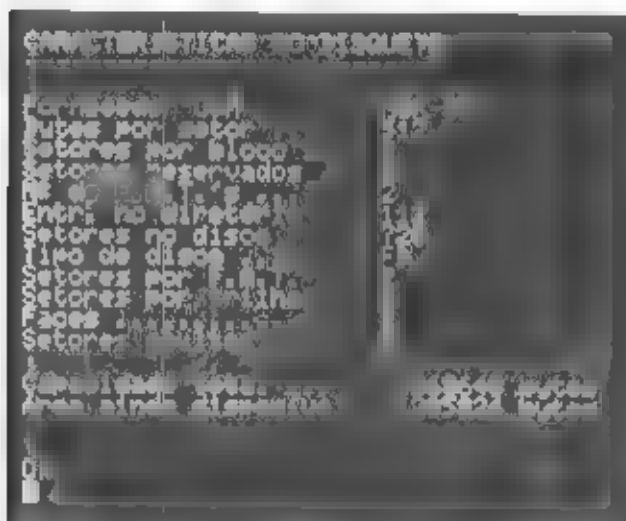
```

3410 PRINT ' bytes"
3420 PRINT "Área ainda disponível:"
3430 D DSKF(0):S=FNB(17):LOCATE 22,17
3440 PRINT USING "#####";D*S*85;
3450 PRINT ' bytes":PRINT:LOCATE 8,21,0

```

DISK  
EXT  
F603  
143  
1070

TOTAL = 1090



## B.4 - MAPA DE DISCOS

O programa apresentado a seguir lê a FAT do disco presente no drive A e apresenta na tela o conteúdo de cada uma de suas posições. Em seguida o diretório do disco é lido e cada um de seus arquivos são mostrados na tela com todos os seus parâmetros especificados incluindo os blocos do disco em que ele está armazenado.

Após digitar e conferir o programa grave-o no formato ASCII para posterior utilização com o comando MERGE.

Caso um arquivo tenha sido apagado com o comando KILL do DISK BASIC ou ERASE ou DEL do MSDOS o programa permitirá a sua recuperação mas APENAS NO DIRETÓRIO. Se o arquivo apagado ocupar menos que um bloco do disco e ele será automaticamente recuperado também na FAT mas se ele ocupar mais de um bloco apesar de ser recuperado no diretório não será mapeado por completo na FAT.

```

4000 REM
4010 REM mapa do disquete
4020 REM
4030 P AY "50M500004H"
4040 SCREEN 0 : W DH 34 : KEY OFF
4050 PRINT SPC(6); ">>> MAPA DO D. 3 1/2"
4060 <<<
4070 PRINT SPC(6); STRING$(4, " ")
4080 PRINT:PRINT:PRINT
4090 PRINT SPC(6); " ) > 1 NOO A.T.A.T. <
4100 <<< LOCATE 0,3,0
4110 GOSUB 8260
4120 FOR F=5 TO 11
4130 LOCATE 0,1,0:PRINT CHR$(27) "I"
4140 PRINT SPC(9); ">>> SET 'P'";
4150 PRINT " NG "HHH";F;
4160 PRINT " <<<"
4170 PRINT SPC(9); "--"
4180 A% = DISK$(0,F)
4190 P = PEEK(&HF351)+256*PEEK(&HF352)
4200 FOR G=0 TO 15
4210 LOCATE 0,5,0:PRINT CHR$(27) "I";
4220 IF INKEY$(CHR$(27)) THEN 1000
4230 PRINT "ARQUIVO";(F-5)*16+G+1
4240 PRINT "-----"
4250 REM
4260 PRINT "NOME"
4270 C% = ""

```

4260	FOR H=0 TO 10	42F
4270	A = P + 32*G + H	BE2
4280	IF PEEK (A) <> 0 THEN 4310	148B
4290	PRINT STRING\$(95,127)	18F2
4300	GOTO 4920	21C9
4310	C\$ = C\$ + CHR\$(PEEK(A))	2349
4320	IF H=7 THEN C\$=C\$+"_"	20B1
4330	NEXT H	220F
4340	PRINT C\$	2130
4350	REM -----	2224
4360	PRINT "ATRIBUTOS :";	6029
4370	AT = PEEK ( P + 32*G + 11 )	11F5
4380	AT\$ = RIGHT\$( "00000000" + BIN\$(AT	8572
4390	),8)	
4400	PRINT " " ; AT\$	BC39
4410	REM -----	9540
4420	PRINT "HORA :";	900A
4430	X = (P+32*G+12)	43B9
4440	X1=PEEK(X)	08F6
4450	X2=PEEK(X+1)	B49A
4460	X1\$ = RIGHT\$( "00000000" + BIN\$(X1),	1191
4470	8)	
4480	X2\$ = RIGHT\$( "00000000" + BIN\$(X2),	BC2E
4490	8)	
4500	H\$ = LEFT\$(X2\$,5)	EE0F
4510	M\$ = RIGHT\$(X2\$,3) + LEFT\$(X1\$,3)	FCF9
4520	H = VAL(" &B" + H\$)	81F
4530	M = VAL(" &B" + M\$)	27C
4540	PRINT " " ; H;M	08C
4550	REM -----	11F5
4560	PRINT "DATA :";	24D8
4570	H2=PEEK(P + 32*G + 24)	2E74
4580	H1=PEEK ( P + 32*G + 25)	BE76
4590	H1\$ = RIGHT\$( "00000000" + BIN\$(H1),	51C3
4600	8)	
4610	H2\$ = RIGHT\$( "00000000" + BIN\$(H2),	5011
4620	8)	
4630	D = VAL(" &B" + RIGHT\$(H2\$,5))	01F5
4640	M = VAL(" &B" + RIGHT\$(H1\$,1) + LEFT\$(	847B
4650	H2\$,3))	
4660	A = 1980 + VAL(" &B" + LEFT\$(H1\$,7))	8023
4670	PRINT D;M;A	940C
4680	REM -----	9F4C
4690	PRINT "10 BLOCO : ";	0E71
4700	H = (P+32*G+26)	B743
4710	H = PEEK(H) + 256*PEEK(H+1)	0623
4720	PB = H	080E
4730	PRINT RIGHT\$( "000" + HEX\$(H),3)	1E23
4740	REM -----	F33E

4690	PRINT "NO DE BYTES: ";	82
4700	H = (P+32*G+28)	141
4710	H = PEEK(H)+256*PEEK(H+1)+	1107
	4096*PEEK(H+2)+65536*PEEK(H+3)	
4720	PRINT RIGHT\$("0000"+HEX\$(H),4)	2519
4730	REM -----	3002
4740	PRINT "MAPA DA FAT: ";	3003
4750	H=P0	3213
4760	IF H>359 THEN 4830	3251
4770	PRINT RIGHT\$("000"+HEX\$(B%(H)),3);	3371
4780	IF H>359 THEN 4830	3470
4790	H=B%(H)	3471
4800	IF H>359 THEN 4830	3473
4810	PRINT RIGHT\$("000"+HEX\$(B%(H)),3);	3487
4820	GOTO 4780	3710
4830	REM -----	3730
4840	PRINT	3731
4850	PRINT	3731
4860	IF LEFT\$(C\$,1) CHR\$(&HEX) THEN	3733
6050	LOCATE 10,23,0	6050
4070	PRINT "DIGITE RETURN: ";	10304
4880	A\$ = INPUT\$(1)	10305
4890	NEXT G	10306
4900	NEXT F	10307
4910	LOCATE 8,23,0	10308
4920	PRINT ">>> TECLA ESPACO <<<";	10309
4930	IF STRIG(0) = 0 THEN 4940 ELSE RUN	10310
4940	REM	10311
8070	REM Recupera deletados	10312
8080	REM -----	10313
8090	PRINT,,"> Arquivo deletado 1 <";	10314
8100	PRINT, "RECUPERAR (S/N) ?";	10315
8110	BEEP : BEEP : BEEP	10316
8120	X = PEEK(&HFCAB) : POKE &HFCAB,1	10317
8130	X\$ = INPUT\$(1) : PRINT X\$	10318
8140	IF X\$<>"S" THEN 8220	10319
8150	PRINT, "10 caracteres do nome: ";	10320
8160	Z\$ = INPUT\$(1) : PRINT Z\$	10321
8170	IF Z\$<"A" OR Z\$>"Z" THEN 8160	10322
8180	POKE (A 10),ASC(Z\$)	10323
8190	DSK0\$ 0,F	10324
8200	GOTO 8230	10325
8210	IF X\$<>"N" THEN 8110	10326
8220	POKE &HFCAB,X : PRINT : PRINT	10327
8230	RETURN	10328
8240	REM	10329

8260	REM leitor de F.A.T.	A8C5
8270	REM ---	B6C0
8280	AZ(0) BZ(0):FRASE AZ,BZ	C428
8290	DIM AZ(539),BZ(359)	D08A
8300	A% ORKIS(0,1)	D817
8310	P = PEEK(&HF351)+256*PEEK(&HF352)	E683
8320	FOR F=0 TO 511	EE5C
8330	AZ(F) = PEEK(P+F)	F258
8340	NEXT F	F8F9
8350	A% ORKIS(1,2)	FE9D
8360	P = PEEK(&HF351)+256*PEEK(&HF352)	08-
8370	FOR F=0 TO 27	DF-
8380	AZ(F+512) = PEEK(P+F)	F04B
8390	NEXT F	F8F9
8400	G=0	F04B
8410	FOR F=0 TO 549 STEP 3	F8F9
8420	IF INKEY% CHR\$(27) THEN 1000	F04B
8430	BZ(G) = AZ(F)+256*(AZ(F+1) AND &HF	4226
8440	BZ(G+1) (AZ(F+1) AND &HF0)/16 +	5823
	AZ(F+2)*16	
8450	G=G+2	F8F9
8460	NEXT F	F04B
8470	FOR F=1 TO 360	F04B
8480	IF INKEY% CHR\$(27) THEN 1000	F8F9
8490	PRINT RIGHT\$("000"+HEX\$(F-1),3);"	F04B
8500	W":RIGHT\$("000"+HEX\$(BZ(F-1)),3);"	F04B
8510	IF F/72 (>) F\72 THEN 8550	C089
8520	LOCATE 10,23,1	C8F0
8530	PRINT "DIGITE RETURN:";	C089
8540	A% INPUT\$(1)	C8F0
8550	LOCATE 0,3,0	C089
8560	NEXT F	C8F0
8570	PRINT CHR\$(27);"J"	C8F0
8580	RETURN	C8F0

TOTAL = C8F2



## B.5 - LEITOR DE SETORES

O programa a seguir gera na tela um "DUMP" do setor de disco especificado pelo usuário. Após digitá-lo e conferi-lo, grave-o em formato ASCII para posteror utilização com o comando MERGE.

```

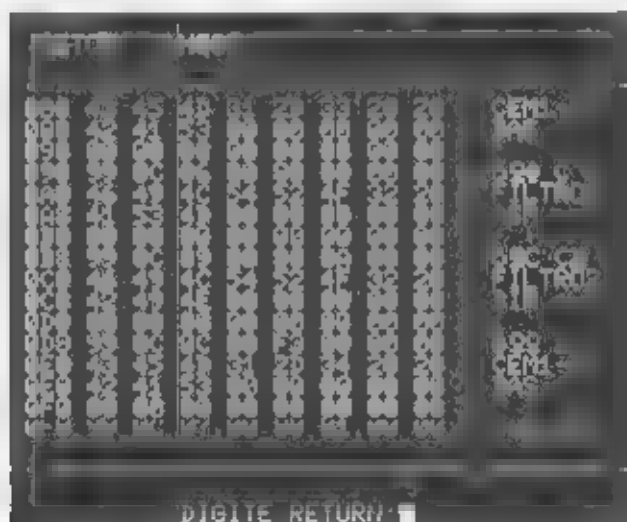
100 CLEAR 1000,&HC000
105 ONERROR GOTO 100
110 RESTORE 150
120 FOR FZ=&HC000 TO &HC02D
130 READ X% : POKE FZ,VAL("&H"+X%)
140 NEXT FZ : DEFUSR0 &HC000
150 DATA 21,00,00,11,2E,C0,01,C0
160 DATA 03,C0,59,00,01,58,02,21
170 DATA F6,C0,11,CE,C0,ED,B0,06
180 DATA 28,21,4E,C3,36,20,23,10
190 DATA FH,21,2E,C0,11,00,00,01
200 DATA C0,03,C0,5C,00,C9,F3,21
210 DATA 32,45,4E,41,54,4F,20,44
220 DATA 41,20,53,49,4C,56,41,20
230 DATA 4F,4C,49,56,45,49,52,41
5000 REM
5010 REM      dump de setores
5020 REM -----
5030 PLAY"50M500005G4"
5040 SCREEN 0 : WIDTH 39
5050 X%=DSKIS(0,0)
5060 E=PEEK(&HF351)+256*PEEK(&HF352)
5070 NS=256*PEEK(E+20)+PEEK(E+20+1)-1
5080 S$=""
5090 PRINT "ENTRE O SETOR : &H ";
5100 PRINT CHR$(8);CHR$(8);
5110 X%=INPUT$(1) : PRINT X%;
5120 IF X% CHR$(8) THEN 5000
5130 S$ S$+X%
5140 IF LEN(S$)<3 THEN 5110
5150 S$ RIGHT$("00"+S$,3)
5160 S=VAL("&H"+S$)
5170 IF S<0 OR S>NS THEN 5000
5180 CLS
5190 PRINT : PRINT " SETOR : &H";S$
5200 PRINT : PRINT STRING$(39,"-")
5210 LOCATE 0,21:PRINT STRING$(39,"-")
5220 X%=DSKIS(0,S)
5230 EN=PEEK(&HF351)+256*PEEK(&HF352)
5240 FOR F=0 TO 63
5250     IF INKEY$=CHR$(27) THEN 1000
5260     IF F/16<>F\16 THEN 5320

```

24-  
 27-  
 28-  
 33-  
 1310  
 1354  
 2400  
 2421  
 2432  
 2443  
 2477  
 2486  
 2495  
 2503  
 2510  
 2519  
 2528  
 2537  
 2546  
 2555  
 2564  
 2573  
 2582  
 2591  
 2600  
 2609  
 2618  
 2627  
 2636  
 2645  
 2654  
 2663  
 2672  
 2681  
 2690  
 2699  
 2708  
 2717  
 2726  
 2735  
 2744  
 2753  
 2762  
 2771  
 2780  
 2789  
 2798  
 2807  
 2816  
 2825  
 2834  
 2843  
 2852  
 2861  
 2870  
 2879  
 2888  
 2897  
 2906  
 2915  
 2924  
 2933  
 2942  
 2951  
 2960  
 2969  
 2978  
 2987  
 2996  
 3005  
 3014  
 3023  
 3032  
 3041  
 3050  
 3059  
 3068  
 3077  
 3086  
 3095  
 3104  
 3113  
 3122  
 3131  
 3140  
 3149  
 3158  
 3167  
 3176  
 3185  
 3194  
 3203  
 3212  
 3221  
 3230  
 3239  
 3248  
 3257  
 3266  
 3275  
 3284  
 3293  
 3302  
 3311  
 3320  
 3329  
 3338  
 3347  
 3356  
 3365  
 3374  
 3383  
 3392  
 3401  
 3410  
 3419  
 3428  
 3437  
 3446  
 3455  
 3464  
 3473  
 3482  
 3491  
 3500  
 3509  
 3518  
 3527  
 3536  
 3545  
 3554  
 3563  
 3572  
 3581  
 3590  
 3599  
 3608  
 3617  
 3626  
 3635  
 3644  
 3653  
 3662  
 3671  
 3680  
 3689  
 3698  
 3707  
 3716  
 3725  
 3734  
 3743  
 3752  
 3761  
 3770  
 3779  
 3788  
 3797  
 3806  
 3815  
 3824  
 3833  
 3842  
 3851  
 3860  
 3869  
 3878  
 3887  
 3896  
 3905  
 3914  
 3923  
 3932  
 3941  
 3950  
 3959  
 3968  
 3977  
 3986  
 3995  
 4004  
 4013  
 4022  
 4031  
 4040  
 4049  
 4058  
 4067  
 4076  
 4085  
 4094  
 4103  
 4112  
 4121  
 4130  
 4139  
 4148  
 4157  
 4166  
 4175  
 4184  
 4193  
 4202  
 4211  
 4220  
 4229  
 4238  
 4247  
 4256  
 4265  
 4274  
 4283  
 4292  
 4301  
 4310  
 4319  
 4328  
 4337  
 4346  
 4355  
 4364  
 4373  
 4382  
 4391  
 4400  
 4409  
 4418  
 4427  
 4436  
 4445  
 4454  
 4463  
 4472  
 4481  
 4490  
 4500  
 4509  
 4518  
 4527  
 4536  
 4545  
 4554  
 4563  
 4572  
 4581  
 4590  
 4600  
 4609  
 4618  
 4627  
 4636  
 4645  
 4654  
 4663  
 4672  
 4681  
 4690  
 4700  
 4709  
 4718  
 4727  
 4736  
 4745  
 4754  
 4763  
 4772  
 4781  
 4790  
 4800  
 4809  
 4818  
 4827  
 4836  
 4845  
 4854  
 4863  
 4872  
 4881  
 4890  
 4900  
 4909  
 4918  
 4927  
 4936  
 4945  
 4954  
 4963  
 4972  
 4981  
 4990  
 5000



5270	LOCATE 10,23,0	FFFF
5280	PRINT "DIGITE RETURN:";	FFFF
5290	Y%=INPUT\$(1)	902F
5300	LOCATE 10,23,0	96CE
5310	PRINT " ";	A09C
5320	X=USR(0)	A68F
5330	LOCATE 0,19,0	A07F
5340	PRINT RIGHT\$("00"+HEX\$(B*F),3);	B487
5350	FOR G=0 TO 7	C7D0
5360	X=PEEK(EN+B*F+G)	C940
5370	Y%=RIGHT\$("0"+HEX\$(X),2)	D09F
5380	PRINT " ";Y%;	E7E4
5390	NEXT G	E088
5400	PRINT " 1 ";	F303
5410	FOR G=0 TO 7	F2E
5420	X=PEEK(EN+B*F+G)	F707
5430	Y%=CHR\$(X)	37
5440	IF X>31 THEN 5460	B07
5450	Y%=CHR\$(1)+CHR\$(64+X)	F304
5460	PRINT Y%;	F774
5470	NEXT G	F002
5480	PRINT	0088
5490	NEXT F	D154



**B.B - LEITOR DE ARQUIVOS**

O programa a seguir gera na tela um "DUMP" do arquivo especificado pelo usuário.

Note que as linhas entre 100 e 230 são idênticas às do programa leitor de setores.

Após digitar e conferir o programa, grave-o em formato ASCII para posterior utilização com o comando MERGE.

```

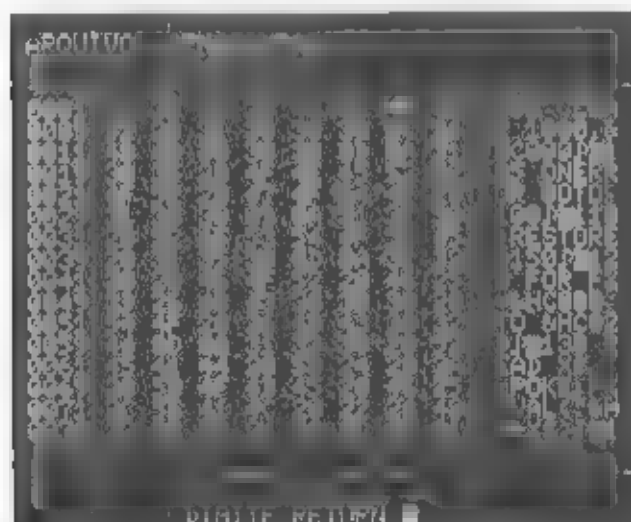
100 CLEAR 1000,&HC000,&HC000
105 ONERROR GOTO 100
110 RESTORE 150
120 FOR FZ=&HC000 TO &HC02D
130 READ X% : POKE FZ,VAL("&H"+X%)
140 NEXT FZ : DEFUSR0=&HC000
150 DATA 21,00,00,11,2E,C0,01,C0
160 DATA 03,CD,59,00,01,58,02,21
170 DATA F6,C0,11,CE,C0,ED,B0,06
180 DATA 28,21,4E,C3,36,20,23,10
190 DATA 18,21,2E,C0,11,00,00,01
200 DATA C0,03,CD,5C,00,C9,F3,21
210 DATA 52,45,4E,41,54,4F,20,44
220 DATA 41,20,53,49,4C,56,41,20
230 DATA 4F,4C,49,56,45,49,52,41
6000 REM
6010 REM      dump de arquivos
6020 REM      -----
6030 PLAY"$OM500003DH"
6040 SCREEN 0 : WIDTH 39
6050 PRINT "ARQUIVOS:" : PRINT : FILES
6060 PRINT : PRINT
6070 PRINT "Entre o nome do arquivo:";
6080 INPUT X%
6090 IF X%="" THEN 1000
6100 CLS
6110 PRINT : PRINT "ARQUIVO :";X%
6120 PRINT : PRINT STRING$(39," ")
6130 LOCATE 0,21:PRINT STRING$(39," ")
6140 OPEN X% AS #1 LEN=1
6150 FIELD #1,1 AS Y%
6160 LF=INT(LOF(1)/8)+1
6170 FOR F=0 TO LF
6180   IF INKEY$=CHR$(27) THEN 1000
6190   IF F/16<>F\16 THEN 6250
6200   LOCATE 10,23,0
6210   PRINT "DIGITE RETURN:";
6220   X%=INPUT$(1)
6230   LOCATE 10,23,0

```

325  
677  
895  
E25  
1-933  
F025  
F052  
G-55  
G73F  
H242  
D-12  
H446  
H406  
8-765  
S6A9  
S607  
S0-73  
H0 14  
H554  
H767  
O477  
O4 13  
P25A  
E617  
F77A  
F0A0  
F35C  
H3  
S51C  
H5  
H00  
H3 11  
H071  
2754  
S903  
G050  
G-69  
S657  
S697

6240	PRINT "	7	1024
6250	X=USR0(0)		2310
6260	LOCATE 0,19,0		2582
6270	PRINT RIGHT\$("000"+HEX\$(1+B*F),4)		1502
?			
6280	X\$=""		5150
6290	FOR G=F*B+1 TO F*B+B		5402
6300	IF G<=LOF(1) THEN 6340		5507
6310	Z\$=""		4440
6320	X\$=X\$+" "		5042
6330	GOTO 6390		5742
6340	GET M1,G		5804
6350	X=ASC(Y\$)		6502
6360	IF X>31 THEN X\$=X\$+CHR\$(X)		0502
6370	IF X<32 THEN X\$=X\$+CHR\$(1)+CHR\$(		1412
(64+X)			
6380	Z\$ RIGHT\$("0"+HEX\$(X),2)		1352
6390	PRINT " ",Z\$;		1752
6400	NEXT G		0
6410	PRINT " ";X\$		517
6420	NEXT F		402

TOTAL	*	415
-------	---	-----



## 6.7 - PROGRAMADOR DE FUNÇÕES

O programa apresentado a seguir permite a programação das teclas de função de forma bastante simples. Após digitar-lo e conferi-lo, grave-o com o nome "FUNKEY.BAS".

O uso deste programa é mais indicado para o MSXDOS. Nesse caso, deve-se alterar o "END" da linha 640 para "\_SYSTEM" e gerar um arquivo BAT com o comando "BAS C FUNKEY.BAS". Assim, estando em DOS, bastará executar o arquivo BAT para poder programar as teclas de função.

```
100 CLEAR 500 : DEFINT A-Z
110 SCREEN 0,,0 : WIDTH 38 : KEY OFF
120 FOR F=1 TO 4 : KEY(F) ON : NEXT F
130 ON KEY GOSUB 490,650,760,790
140 PRINT "REDEFINIDOR DE FUNCOES";
150 PRINT " - EDITORA ALEPH";
160 PRINTSTRING$(38,"");
170 PRINT : PRINT : PRINT : PRINT
180 PRINT CHR$(27);"y4"
190 FOR F=1 TO 10
200     PRINT SPC(7);
210     PRINT "F";USING"##";F;
220     PRINT " --->[                ]"
230 NEXT F
240 PRINT : PRINT
250 PRINT SPC(11); "F1 - VOLTA AO DOS"
260 PRINT SPC(11); "F2 - REINICIALIZA"
270 PRINT SPC(11); "F3 - DEFAULT BASIC"
280 PRINT SPC(11); "F4 - LIMPA TEXTOS"
290 X=16 : Y=7
300 LOCATE X,Y,1
310 AS INKEY$
320 IF AS<CHR$(28) THEN GOTO 350
330 IF AS>CHR$(31) THEN GOTO 350
340 AS=""
350 A=STICK(0)
360 IF A>0 AND A<3 OR A=8 THEN Y=Y-1
370 IF Y=6 THEN Y=16
380 IF A>3 AND A<7 THEN Y=Y+1
390 IF Y=17 THEN Y=7
400 IF A>1 AND A<5 THEN X=X+1
410 IF X>30 THEN X=16
420 IF A>5 THEN X=X-1
430 IF X=15 THEN X=30
440 IF AS="" THEN GOTO 300
450 LOCATE,,0
```

1000  
1010  
1020  
1030  
1040  
1050  
1060  
1070  
1080  
1090  
1100  
1110  
1120  
1130  
1140  
1150  
1160  
1170  
1180  
1190  
1200  
1210  
1220  
1230  
1240  
1250  
1260  
1270  
1280  
1290  
1300  
1310  
1320  
1330  
1340  
1350  
1360  
1370  
1380  
1390  
1400  
1410  
1420  
1430  
1440  
1450  
1460  
1470  
1480  
1490  
1500  
1510  
1520  
1530  
1540  
1550  
1560  
1570  
1580  
1590  
1600  
1610  
1620  
1630  
1640  
1650  
1660  
1670  
1680  
1690  
1700  
1710  
1720  
1730  
1740  
1750  
1760  
1770  
1780  
1790  
1800  
1810  
1820  
1830  
1840  
1850  
1860  
1870  
1880  
1890  
1900  
1910  
1920  
1930  
1940  
1950  
1960  
1970  
1980  
1990  
2000  
2010  
2020  
2030  
2040  
2050  
2060  
2070  
2080  
2090  
2100  
2110  
2120  
2130  
2140  
2150  
2160  
2170  
2180  
2190  
2200  
2210  
2220  
2230  
2240  
2250  
2260  
2270  
2280  
2290  
2300  
2310  
2320  
2330  
2340  
2350  
2360  
2370  
2380  
2390  
2400  
2410  
2420  
2430  
2440  
2450  
2460  
2470  
2480  
2490  
2500  
2510  
2520  
2530  
2540  
2550  
2560  
2570  
2580  
2590  
2600  
2610  
2620  
2630  
2640  
2650  
2660  
2670  
2680  
2690  
2700  
2710  
2720  
2730  
2740  
2750  
2760  
2770  
2780  
2790  
2800  
2810  
2820  
2830  
2840  
2850  
2860  
2870  
2880  
2890  
2900  
2910  
2920  
2930  
2940  
2950  
2960  
2970  
2980  
2990  
3000  
3010  
3020  
3030  
3040  
3050  
3060  
3070  
3080  
3090  
3100  
3110  
3120  
3130  
3140  
3150  
3160  
3170  
3180  
3190  
3200  
3210  
3220  
3230  
3240  
3250  
3260  
3270  
3280  
3290  
3300  
3310  
3320  
3330  
3340  
3350  
3360  
3370  
3380  
3390  
3400  
3410  
3420  
3430  
3440  
3450  
3460  
3470  
3480  
3490  
3500  
3510  
3520  
3530  
3540  
3550  
3560  
3570  
3580  
3590  
3600  
3610  
3620  
3630  
3640  
3650  
3660  
3670  
3680  
3690  
3700  
3710  
3720  
3730  
3740  
3750  
3760  
3770  
3780  
3790  
3800  
3810  
3820  
3830  
3840  
3850  
3860  
3870  
3880  
3890  
3900  
3910  
3920  
3930  
3940  
3950  
3960  
3970  
3980  
3990  
4000  
4010  
4020  
4030  
4040  
4050  
4060  
4070  
4080  
4090  
4100  
4110  
4120  
4130  
4140  
4150  
4160  
4170  
4180  
4190  
4200  
4210  
4220  
4230  
4240  
4250  
4260  
4270  
4280  
4290  
4300  
4310  
4320  
4330  
4340  
4350  
4360  
4370  
4380  
4390  
4400  
4410  
4420  
4430  
4440  
4450  
4460  
4470  
4480  
4490  
4500  
4510  
4520  
4530  
4540  
4550  
4560  
4570  
4580  
4590  
4600  
4610  
4620  
4630  
4640  
4650  
4660  
4670  
4680  
4690  
4700  
4710  
4720  
4730  
4740  
4750  
4760  
4770  
4780  
4790  
4800  
4810  
4820  
4830  
4840  
4850  
4860  
4870  
4880  
4890  
4900  
4910  
4920  
4930  
4940  
4950  
4960  
4970  
4980  
4990  
5000  
5010  
5020  
5030  
5040  
5050  
5060  
5070  
5080  
5090  
5100  
5110  
5120  
5130  
5140  
5150  
5160  
5170  
5180  
5190  
5200  
5210  
5220  
5230  
5240  
5250  
5260  
5270  
5280  
5290  
5300  
5310  
5320  
5330  
5340  
5350  
5360  
5370  
5380  
5390  
5400  
5410  
5420  
5430  
5440  
5450  
5460  
5470  
5480  
5490  
5500  
5510  
5520  
5530  
5540  
5550  
5560  
5570  
5580  
5590  
5600  
5610  
5620  
5630  
5640  
5650  
5660  
5670  
5680  
5690  
5700  
5710  
5720  
5730  
5740  
5750  
5760  
5770  
5780  
5790  
5800  
5810  
5820  
5830  
5840  
5850  
5860  
5870  
5880  
5890  
5900  
5910  
5920  
5930  
5940  
5950  
5960  
5970  
5980  
5990  
6000  
6010  
6020  
6030  
6040  
6050  
6060  
6070  
6080  
6090  
6100  
6110  
6120  
6130  
6140  
6150  
6160  
6170  
6180  
6190  
6200  
6210  
6220  
6230  
6240  
6250  
6260  
6270  
6280  
6290  
6300  
6310  
6320  
6330  
6340  
6350  
6360  
6370  
6380  
6390  
6400  
6410  
6420  
6430  
6440  
6450  
6460  
6470  
6480  
6490  
6500  
6510  
6520  
6530  
6540  
6550  
6560  
6570  
6580  
6590  
6600  
6610  
6620  
6630  
6640  
6650  
6660  
6670  
6680  
6690  
6700  
6710  
6720  
6730  
6740  
6750  
6760  
6770  
6780  
6790  
6800  
6810  
6820  
6830  
6840  
6850  
6860  
6870  
6880  
6890  
6900  
6910  
6920  
6930  
6940  
6950  
6960  
6970  
6980  
6990  
7000  
7010  
7020  
7030  
7040  
7050  
7060  
7070  
7080  
7090  
7100  
7110  
7120  
7130  
7140  
7150  
7160  
7170  
7180  
7190  
7200  
7210  
7220  
7230  
7240  
7250  
7260  
7270  
7280  
7290  
7300  
7310  
7320  
7330  
7340  
7350  
7360  
7370  
7380  
7390  
7400  
7410  
7420  
7430  
7440  
7450  
7460  
7470  
7480  
7490  
7500  
7510  
7520  
7530  
7540  
7550  
7560  
7570  
7580  
7590  
7600  
7610  
7620  
7630  
7640  
7650  
7660  
7670  
7680  
7690  
7700  
7710  
7720  
7730  
7740  
7750  
7760  
7770  
7780  
7790  
7800  
7810  
7820  
7830  
7840  
7850  
7860  
7870  
7880  
7890  
7900  
7910  
7920  
7930  
7940  
7950  
7960  
7970  
7980  
7990  
8000  
8010  
8020  
8030  
8040  
8050  
8060  
8070  
8080  
8090  
8100  
8110  
8120  
8130  
8140  
8150  
8160  
8170  
8180  
8190  
8200  
8210  
8220  
8230  
8240  
8250  
8260  
8270  
8280  
8290  
8300  
8310  
8320  
8330  
8340  
8350  
8360  
8370  
8380  
8390  
8400  
8410  
8420  
8430  
8440  
8450  
8460  
8470  
8480  
8490  
8500  
8510  
8520  
8530  
8540  
8550  
8560  
8570  
8580  
8590  
8600  
8610  
8620  
8630  
8640  
8650  
8660  
8670  
8680  
8690  
8700  
8710  
8720  
8730  
8740  
8750  
8760  
8770  
8780  
8790  
8800  
8810  
8820  
8830  
8840  
8850  
8860  
8870  
8880  
8890  
8900  
8910  
8920  
8930  
8940  
8950  
8960  
8970  
8980  
8990  
9000  
9010  
9020  
9030  
9040  
9050  
9060  
9070  
9080  
9090  
9100  
9110  
9120  
9130  
9140  
9150  
9160  
9170  
9180  
9190  
9200  
9210  
9220  
9230  
9240  
9250  
9260  
9270  
9280  
9290  
9300  
9310  
9320  
9330  
9340  
9350  
9360  
9370  
9380  
9390  
9400  
9410  
9420  
9430  
9440  
9450  
9460  
9470  
9480  
9490  
9500  
9510  
9520  
9530  
9540  
9550  
9560  
9570  
9580  
9590  
9600  
9610  
9620  
9630  
9640  
9650  
9660  
9670  
9680  
9690  
9700  
9710  
9720  
9730  
9740  
9750  
9760  
9770  
9780  
9790  
9800  
9810  
9820  
9830  
9840  
9850  
9860  
9870  
9880  
9890  
9900  
9910  
9920  
9930  
9940  
9950  
9960  
9970  
9980  
9990  
10000  
10010  
10020  
10030  
10040  
10050  
10060  
10070  
10080  
10090  
10100  
10110  
10120  
10130  
10140  
10150  
10160  
10170  
10180  
10190  
10200  
10210  
10220  
10230  
10240  
10250  
10260  
10270  
10280  
10290  
10300  
10310  
10320  
10330  
10340  
10350  
10360  
10370  
10380  
10390  
10400  
10410  
10420  
10430  
10440  
10450  
10460  
10470  
10480  
10490  
10500  
10510  
10520  
10530  
10540  
10550  
10560  
10570  
10580  
10590  
10600  
10610  
10620  
10630  
10640  
10650  
10660  
10670  
10680  
10690  
10700  
10710  
10720  
10730  
10740  
10750  
10760  
10770  
10780  
10790  
10800  
10810  
10820  
10830  
10840  
10850  
10860  
10870  
10880  
10890  
10900  
10910  
10920  
10930  
10940  
10950  
10960  
10970  
10980  
10990  
11000  
11010  
11020  
11030  
11040  
11050  
11060  
11070  
11080  
11090  
11100  
11110  
11120  
11130  
11140  
11150  
11160  
11170  
11180  
11190  
11200  
11210  
11220  
11230  
11240  
11250  
11260  
11270  
11280  
11290  
11300  
11310  
11320  
11330  
11340  
11350  
11360  
11370  
11380  
11390  
11400  
11410  
11420  
11430  
11440  
11450  
11460  
11470  
11480  
11490  
11500  
11510  
11520  
11530  
11540  
11550  
11560  
11570  
11580  
11590  
11600  
11610  
11620  
11630  
11640  
11650  
11660  
11670  
11680  
11690  
11700  
11710  
11720  
11730  
11740  
11750  
11760  
11770  
11780  
11790  
11800  
11810  
11820  
11830  
11840  
11850  
11860  
11870  
11880  
11890  
11900  
11910  
11920  
11930  
11940  
11950  
11960  
11970  
11980  
11990  
12000  
12010  
12020  
12030  
12040  
12050  
12060  
12070  
12080  
12090  
12100  
12110  
12120  
12130  
12140  
12150  
12160  
12170  
12180  
12190  
12200  
12210  
12220  
12230  
12240  
12250  
12260  
12270  
12280  
12290  
12300  
12310  
12320  
12330  
12340  
12350  
12360  
12370  
12380  
12390  
12400  
12410  
12420  
12430  
12440  
12450  
12460  
12470  
12480  
12490  
12500  
12510  
12520  
12530  
12540  
12550  
12560  
12570  
12580  
12590  
12600  
12610  
12620  
12630  
12640  
12650  
12660  
12670  
12680  
12690  
12700  
12710  
12720  
12730  
12740  
12750  
12760  
12770  
12780  
12790  
12800  
12810  
12820  
12830  
12840  
12850  
12860  
12870  
12880  
12890  
12900  
12910  
12920  
12930  
12940  
12950  
12960  
12970  
12980  
12990  
13000  
13010  
13020  
13030  
13040  
13050  
13060  
13070  
13080  
13090  
13100  
13110  
13120  
13130  
13140  
13150  
13160  
13170  
13180  
13190  
13200  
13210  
13220  
13230  
13240  
13250  
13260  
13270  
13280  
13290  
13300  
13310  
13320  
13330  
13340  
13350  
13360  
13370  
13380  
13390  
13400  
13410  
13420  
13430  
13440  
13450  
13460  
13470  
13480  
13490  
13500  
13510  
13520  
13530  
13540  
13550  
13560  
13570  
13580  
13590  
13600  
13610  
13620  
13630  
13640  
13650  
13660  
13670  
13680  
13690  
13700  
13710  
13720  
13730  
13740  
13750  
13760  
13770  
13780  
13790  
13800  
13810  
13820  
13830  
13840  
13850  
13860  
13870  
13880  
13890  
13900  
13910  
13920  
13930  
13940  
13950  
13960  
13970  
13980  
13990  
14000  
14010  
14020  
14030  
14040  
14050  
14060  
14070  
14080  
14090  
14100  
14110  
14120  
14130  
14140  
14150  
14160  
14170  
14180  
14190  
14200  
14210  
14220  
14230  
14240  
14250  
14260  
14270  
14280  
14290  
14300  
14310  
14320  
14330  
14340  
14350  
14360  
14370  
14380  
14390  
14400  
14410  
14420  
14430  
14440  
14450  
14460  
14470  
14480  
14490  
14500  
14510  
14520  
14530  
14540  
14550  
14560  
14570  
14580  
14590  
14600  
14610  
14620  
14630  
14640  
14650  
14660  
14670  
14680  
14690  
14700  
14710  
14720  
14730  
14740  
14750  
14760  
14770  
14780  
14790  
14800  
14810  
14820  
14830  
14840  
14850  
14860  
14870  
14880  
14890  
14900  
14910  
14920  
14930  
14940  
14950  
14960  
14970  
14980  
14990  
15000  
15010  
15020  
15030  
15040  
15050  
15060  
15070  
15080  
15090  
15100  
15110  
15120  
15130  
15140  
15150  
15160  
15170  
15180  
15190  
15200  
15210  
15220  
15230  
15240  
15250  
15260  
15270  
15280  
15290  
15300  
15310  
15320  
15330  
15340  
15350  
15360  
15370  
15380  
15390  
15400  
15410  
15420  
15430  
15440  
15450  
15460  
15470  
15480  
15490  
15500  
15510  
15520  
15530  
15540  
15550  
15560  
15570  
15580  
15590  
15600  
15610  
15620  
15630  
15640  
15650  
15660  
15670  
15680  
15690  
15700  
15710  
15720  
15730  
15740  
15750  
15760  
15770  
15780  
15790  
15800  
15810  
15820  
15830  
15840  
15850  
15860  
15870  
15880  
15890  
15900  
15910  
15920  
15930  
15940  
15950  
15960  
15970  
15980  
15990  
16000  
16010  
16020  
16030  
16040  
16050  
16060  
16070  
16080  
16090  
16100  
16110  
16120  
16130  
16140  
16150  
16160  
16170  
16180  
16190  
16200  
16210  
16220  
16230  
16240  
16250  
16260  
16270  
16280  
16290  
16300  
16310  
16320  
16330  
16340  
16350  
16360  
16370  
16380  
16390  
16400  
16410  
16420  
16430  
16440  
16450  
16460  
16470  
16480  
16490  
16500  
16510  
16520  
16530  
16540  
16550  
16560  
16570  
16580  
16590  
16600  
16610  
16620  
16630  
16640  
16650  
16660  
16670  
16680  
16690  
16700  
16710  
16720  
16730  
16740  
16750  
16760  
16770  
16780  
16790  
16800  
16810  
16820  
16830  
16840  
16850  
16860  
16870  
16880  
16890  
16900  
16910  
16920  
16930  
16940  
16950  
16960  
16970  
16980  
16990  
17000  
17010  
17020  
17030  
17040  
17050  
17060  
17070  
17080  
17090  
17100  
17110  
17120  
17130  
17140  
17150  
17160  
17170  
17180  
17190  
17200  
17210  
17220  
17230  
17240  
17250  
17260  
17270  
17280  
17290  
17300  
17310  
17320  
17330  
17340  
17350  
17360  
17370  
17380  
17390  
17400  
17410  
17420  
17430  
17440  
17450  
17460  
17470  
17480  
17490  
17500  
17510  
17520  
17530  
17540  
17550  
17560  
17570  
17580  
17590  
17600  
17610  
17620  
17630  
17640  
17650  
17660  
17670  
17680  
17690  
17700  
17710  
17720  
17730  
17740  
17750  
17760  
17770  
17780  
17790  
17800  
17810  
17820  
17830  
17840  
17850  
17860  
17870  
17880  
17890  
17900  
17910  
17920  
17930  
17940  
17950  
17960  
17970  
17980  
17990  
18000  
18010  
18020  
18030  
18040  
18050  
18060  
18070  
18080  
18090  
18100  
18110  
18120  
18130  
18140  
18150  
18160  
18170  
18180  
18190  
18200  
18210  
18220  
18230  
18240  
18250  
18260  
18270  
18280  
18290  
18300  
18310  
18320  
18330  
18340  
18350  
18360  
18370  
18380  
18390  
18400  
18410  
18420  
18430  
18440  
18450  
18460  
18470  
18480  
18490  
18500  
18510  
18520  
18530  
18540  
18550  
18560  
18570  
18580  
18590  
18600  
18610  
18620  
18630  
18640  
18650  
18660  
18670  
18680  
18690  
18700  
18710  
18720  
18730  
18740  
18750  
18760  
18770  
18780  
18790  
18800  
18810  
18820  
18830  
1

```

460 VPOKE 40*Y+X+1,ASC(A$)
470 IF X<30 THEN X=X+1
480 GOTO 300
490 BEEP : BEEP : BEEP
500 FOR F=1 TO 10
510     LOCATE 0,0,0
520     A$="" : FL=0
530     X=(F+6)*40+17
540     FOR Y=X+14 TO X STEP -1
550         IF FL=1 THEN 580
560         IF VPEEK(Y)=32 THEN 590
570         FL=1
580         A$=CHR$(VPEEK(Y))+A$
590     NEXT Y
600     KEY F,A$
610 NEXT F
620 SCREEN 0 : WIDTH 40 : KEY 0
630 FOR F=1 TO 4 : KEY(F) OFF : NEXT F
640 PRINT CHR$(27);"x4" : END
650 BEEP : BEEP : BEEP
660 GOSUB 790 : LOCATE,,0
670 FOR F=0 TO 9
680     FOR G=0 TO 15
690         A=PEEK(&HFB7F+16*F+G)
700         IF A=0 THEN 720
710         VPOKE40*(F+7)+17+G,A
720     NEXT G
730 NEXT F
740 LOCATE,,1
750 RETURN
760 BEEP : BEEP : BEEP
770 DEFUSR &H3E : S=USR(0)
780 GOSUB 650 : RETURN
790 BEEP : BEEP : BEEP
800 LOCATE,,0 : PRINT " ",CHR$(S)
810 FOR F=0 TO 9
820     FOR G=0 TO 14
830         VPOKE40*(F+7)+17+G,32
840     NEXT G
850 NEXT F
860 LOCATE,,1
870 RETURN

```

460  
470  
480  
490  
500  
510  
520  
530  
540  
550  
560  
570  
580  
590  
600  
610  
620  
630  
640  
650  
660  
670  
680  
690  
700  
710  
720  
730  
740  
750  
760  
770  
780  
790  
800  
810  
820  
830  
840  
850  
860  
870

460  
470  
480  
490  
500  
510  
520  
530  
540  
550  
560  
570  
580  
590  
600  
610  
620  
630  
640  
650  
660  
670  
680  
690  
700  
710  
720  
730  
740  
750  
760  
770  
780  
790  
800  
810  
820  
830  
840  
850  
860  
870

## 8.8 - CLS EM MSXDOS

O pequeno programa em BASIC listado a seguir gera um arquivo BATCH que limpa a tela ao ser executado a partir do MSXDOS.

Após digitá-lo, confira-lo e gravá-lo, rode-o.

Dessa forma, deverá ter sido gerado no disco presente no drive A um arquivo BAT de nome "CLS.BAT". Esse arquivo deve ser usado a partir do MSXDOS, simplesmente digitando CLS e RETURN.

```
10 OPEN "A:CLS.BAT" AS #1
20 FIELD #1, 2 AS A%
30 LET A%=CHR$(12)+CHR$(26)
40 PUT #1,1
50 END
```

400  
510  
520  
1000  
1541

Após rodar o programa, experimente carregar o MSXDOS e comande:

A>CLS

A tela deverá ser limpa.



## 6.9 - BEEP EM MSXDOS

O programa em BASIC listado a seguir gera um arquivo BATCH que produz um BEEP quando executado a partir do MSXDOS.

Seu uso é semelhante ao do programa CLS BAT gerado na dica 8 C e, portanto ele deve ser executado necessariamente a partir do MSXDOS

```
10 OPEN "A:BEEP.BAT" AS #1
20 PRINT #1 "CLS"
30 PRINT #1 "A%="CHR$(7)+CHR$(26)
40 PUT #1,1
50 END
```

54E  
5844  
F11  
18AF  
1226

TOTAL = 1226

Após rodar o programa carregue o MSXDOS e comande:

A>BEEP

Isso deverá produzir um beep.

## 6.10 - "SWAP" DE ARQUIVOS EM MSXDOS

Estando em MSXDOS, digite a sequência de comandos listada a seguir para produzir um arquivo BATCH capaz de trocar os nomes de dois arquivos quaisquer

```
A>COPY CON A:SWAP.BAT
REN %1 BADA.%%
REN %2 %1
REN BADA.%% %2
^Z
```

Com o arquivo SWAP.BAT já gravado no disco, você poderá usá-lo com a seguinte sintaxe:

A>SWAP arquivo1 arquivo2

Os parâmetros "arquivo1" e "arquivo2" são os nomes dos dois arquivos cujos conteúdos se deseja trocar.



## DICAS DE PROCESSAMENTO

As dicas apresentadas neste capítulo são as mais diversas abordando recursos multíplos das máquinas MSX. Elas não são específicas para nenhum periférico, sendo mais relacionadas com o processamento de outros programas.

7 1	Diferenciando o Expert do Holbit	. 154
7 2	- Escondendo listagens de programas	. 155
7 3	- Desativando todos os comandos	. 157
7 4	- Evitando o "ON STOP GOSUB"	. 158
7 5	- Obtendo o valor de PI	. 159
7 6	- Tabelas de caracteres	. 160
7 7	- Sorteio de palavras da ROM	. 163
7 8	- Sorteio de palavras em linhas "DATA's"	. 164
7 9	- Mapeando linhas REM	. 165
7 A	- Mapeando variáveis	. 166
7 B	- Mapeando linhas de um programa	. 167
7 C	- Gerando linhas DATA .....	. 168
7 D	- Trocando tokens num programa	. 169
7 E	- Aumentando a velocidade de execução ...	. 171
7 F	- "SEARCH" pesquisador de strings	. 173
7 G	- Recuperando programas apagados com NEW	. 175
7 H	- Redefinindo mensagens de erros	. 176
7 I	- Soma sintática	. 178
7 J	- PSEJDO-RAMDISK	. 181
7 K	- Rotacionando caracteres	. 183
7 L	- Rotina de entrada com INKEY\$ ..	. 184
7 M	- Rotina para maiúsculas	. 186
7 N	- Reduzindo a tecla a um caractere	. 187
7 O	- Grandes expoentes	. 188



## 7.1 - DIFERENCIANDO O EXPERT DO HOTBIT

Existem muitas diferenças facilmente identificáveis entre o HOTBIT e o EXPERT. A mais fácil de ser checada por um programa, quer esteja ele em BASIC ou em linguagem de Máquina, é a mensagem de identificação do fabricante. No Expert, a partir do endereço 32513 encontramos na ROM a string "G r a d i e n t e". Basta então checar uma letra dessa mensagem que difere do HOTBIT, por exemplo, a letra "G" (código ASCII 71). O programa a seguir faz exatamente isso.

```
100 SCREEN 0 : WIDTH 39 : PRINT
110 PRINT " Conteúdo da ROM a partir "
120 PRINT " do endereço 32513: ";
130 FOR F=32513 TO 32531
140   PRINT CHR$(PEEK(F));
150 NEXT F
160 X$="é"
170 IF PEEK(32513)()71 THEN X$="não é"
180 PRINT : PRINT : PRINT " ";
190 PRINT "Portanto, ";X$;" um EXPERT!"
200 PRINT : PRINT
```

32513  
32514  
32515  
32516  
32517  
32518  
32519  
32520  
32521  
32522  
32523  
32524  
32525  
32526  
32527  
32528  
32529  
32530  
32531



### BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - páginas 82 e 83

## 7.2 - ESCONDENDO LISTAGENS DE PROGRAMAS

Um programa em BASIC pode estar presente na memória do micro sem que o comando LIST ou LL ST faça sua listagem aparecer na tela ou na impressora.

A seguir são apresentados dois pequenos programas que exemplificam duas das muitas formas disponíveis para se esconder a listagem.

Em ambos os programas os comandos responsáveis pela ocultação das listagens encontram-se na linha 10.

O primeiro programa deve ser rodado ao menos uma vez antes de ser gravado. Feito isso o programa deve ser gravado no formato compactado isto é não pode ser gravado em ASCII. Para fita cassete deve-se usar o comando CSAVE e para disquete deve-se usar o comando SAVE sem a opção "/A".

### PROGRAMA 1

```
10 POKE &H8003, &HFF: POKE &H8004, &HFF
20 FOR F=1 TO 200
30 PRINT "ESTOU AQUI ESCONDIDO !!!"
40 NEXT F
```

1000  
1001  
1002  
1003

INITIAL = 1000

O segundo programa pode ser gravado mesmo sem ter sido rodado pois de qualquer forma, ao ser carregado da fita ou do disquete, sua listagem estará visível. Para que ela se torne necessária é necessário que ele seja rodado ao menos uma vez após ter sido carregado.

### PROGRAMA 2

```
10 POKE &HFFB9, &HD1
20 FOR F=1 TO 200
30 PRINT "ESTOU AQUI ESCONDIDO !!!"
40 NEXT F
```

1000  
1001  
1002  
1003

INITIAL = 1000

Uma variação mais drástica do programa 2 é obtida alterando-se a linha 10 para:

```
10 POKE &HFFB9, &HC3
```

Com isso toda vez que o comando LIST for

executado o micro será automaticamente resettado  
Para desativar essa "trava" basta comandar:

POKE &HFFB7,HC9

Note que apenas o comando LIST foi desabilitado  
Os demais comandos do BASIC continuam funcionando normalmente

Ao invés de resettar o micro podemos simplesmente avtar a stagem Acrescente ao programa 2 as linhas a seguir:

11 POKE &HFF8A,&H72  
12 POKE &HFF8B,&H0



## 7.3 - DESATIVANDO TODOS OS COMANDOS

O hook do "Ok" é chamado pelo Interpretador sempre que o "Ok" vai ser mostrado na tela. Isso pode servir para evitar que sejam usados quaisquer comandos do BASIC MSX. Para produzir um desvio no hook do "Ok" basta usar os seguintes comandos.

```
POKE &HFF07,&HC3  
POKE &HFF08,&H00  
POKE &HFF09,&H00
```

Com isso, sempre que o "Ok" for ser impresso na tela o micro será resettado. Para voltar a operar normalmente basta comandar.

```
POKE &HFF07,&HC9
```

Um exemplo prático dessa "trava" pode ser imaginado ao se interromper a execução de um programa em BASIC com CONTROL + STOP logo a seguir o "Ok" surge na tela, provocando um reset automático do micro.



## 7.4 - EVITANDO O "ON STOP GOSUB"

O MSX permite que se "trave" um programa em BASIC após o início de sua execução com as instruções:

```
STOP ON 5  
ON STOP GOSUB xxxxx
```

Usando essas instruções nas primeiras linhas de um programa após serem executadas e as farão com que sempre que as teclas CONTROL e STOP forem pressionadas conjuntamente a execução seja desviada para a linha de número xxxxx.

Isso pode ser muito útil depois que o programa está pronto mas durante sua elaboração pode trazer transtornos ao programador.

O MSX porém possui o veneno e o antídoto. Com apenas um POKE numa variável do sistema é possível acionar o "warm start" do interpretador BASIC pelo teclado. Imagine que vamos executar um programa que usa as instruções STOP ON e ON STOP GOSUB. Se quisermos torná-lo facilmente interrompível basta comandar:

```
POKE &HFB00,1
```

Isso fará com que o pressionamento conjunto das teclas CONTROL SHIFT LGRA e HGRA no Expert (ou CTRL SHIFT, GRAPH e CODE no MOTO T) interrompa a execução do programa e devolva o comando ao usuário.

Para desligar o acionamento do "warm start" do interpretador basta comandar:

```
POKE &HFB00,0
```

Esse recurso não é exclusivo do BASIC. Mesmo quando um programa em Linguagem de Máquina está sendo executado se o conteúdo de &HFB00 for diferente de 0 a execução pode ser interrompida desde que o programa em L.M. não tenha desabilitado a interrupção!



## 7.5 - OBTENDO O VALOR DE PI

O MSX não possui uma variável reservada para o armazenamento da constante matemática  $\pi$ .

Entretanto, essa constante pode ser facilmente obtida de várias formas diferentes.

A forma mais imediata é usar a expressão:

$4 * \text{ATN}(1)$

Experimente comandar:

`PRINT 4*ATN(1)`

O valor de  $\pi$  surgirá na tela, pois:

$\text{Tan}(\pi/4) = 1$

$\text{Atn}(\text{tan}(\pi/4)) = \text{Atn}(1)$

$\pi/4 = \text{Atn}(1)$

$\pi = 4 * \text{Atn}(1)$

Outra forma de obter o valor dessa constante é buscá-lo na própria ROM do micro. Os dois programas apresentados a seguir fazem exatamente isso.

```
10 REM PIROM I
20 AH = 0
30 FOR F=0 TO 7
40 POKE VARPTR(AH)+F,PEEK(&H2D43+F)
50 NEXT F
60 PI = 2*AH
70 PRINT PI
```

440  
44F  
808  
F6F  
10F4  
1488  
4527

```
10 REM PIROM II
20 FOR F=0 TO 7
30 A% = A% + CHR$(PEEK(&H2D43+F))
40 NEXT F
50 PI = 2*CVI(A%)
60 PRINT PI
```

2F2  
60F  
FF7  
1071  
1678  
1888

PIROM = 3.14159

Após a execução de um desses dois programas o valor de  $\pi$  ficará armazenado na variável `PI`.

## 7.6 - TABELAS DE CARACTERES

O programa apresentado a seguir gera na tela a tabela de caracteres do seu micro. A apresentação é feita em duas partes, a primeira contém os caracteres de 0 a 127 e a segunda os caracteres de 128 a 255.

Além dos próprios caracteres a tabela contém um quadro em branco logo abaixo de cada um deles. Esses quadros estão divididos em três campos e devem ser preenchidos por você após a impressão da tabela em papel. Eles servirão para indicar as teclas a serem pressionadas para que o caractere correspondente seja apresentado na tela. Por exemplo, o quadro preenchido abaixo corresponde ao caractere "9".



O campo principal indica a tecla em que o caractere se encontra. O campo superior à direita, quando preenchido, indica que a tecla SHIFT deve ser pressionada. O campo inferior direito, quando preenchido, indica que a tecla LCRA (ou GRAPH) deve ser pressionada. Portanto, para produzir o caractere "9" deve-se pressionar as teclas,

SHIFT + LCRA + 9                      (Expert)  
OU

SHIFT + GRAPH + 9                      (Hotbit)

O campo inferior direito serve para indicar se a tecla RGRA (ou CODE) deve ser usada. Nesse caso, ele deve ser preenchido.

Caso o caractere não possa ser produzido através do teclado, sugerimos que não se preencha nada no campo reservado para a tecla. Cabe aqui uma crítica aos fabricantes, alguns caracteres não acessáveis pelo

teclado são símbolos reservados de algumas linguagens de programação. O caractere "til", por exemplo, é usado como operador lógico de negação para bits na linguagem C. Isso faz parte do padrão da linguagem e o próprio comp.ador MSX C, da ASC Corporation, o adota. O usuário de um EXPERT ou de um KOTBIT, entretanto, terá que fazer algumas "mágicas" antes de conseguir usar esse operador lógico num programa em C. Outros caracteres como o "6", por exemplo, são repetidos desnecessariamente, uma vez que o caractere maiúsculo e o minúsculo têm o mesmo desenho.

```

100 '
110 ' TABLE DE CARACTERES
120 '
130 SCREEN 2,1
140 OPEN "GRP:" AS #1
150 GOSUB 370 : ' horizontais
160 GOSUB 460 : ' verticais
170 GOSUB 560 : ' tracinhos
180 H = 0
190 GOSUB 250 : ' caracteres
200 AS = INPUT$(1)
210 H = 128
220 GOSUB 250 : ' caracteres
230 AS = INPUT$(1)
240 GOTO 180
250 '
260 ' desenha caracteres
270 '
280 FOR F=H TO H+127
290 L = (F-H)\16
300 C = 4+F-H-16*L
310 LINE(12*C,13+22*L)-
      (12*C+7,20+22*L),1,8F
320 PSET (12*C,13+22*L),1
330 IF F<32 THEN PRINT #1,CHR$(1)+
      CHR$(64+F)
340 IF F>31 THEN PRINT #1,CHR$(F)
350 NEXT F
360 RETURN
370 '
380 ' linhas horizontais
390 '
400 FOR F=0 TO 187 STEP 11
410 IF F/2<>F\2 THEN LINE(33,F)
      -(45,F)
420 LINE(45,F)-(238,F)

```

```

300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```



```

430   IF F/22-F\22 THEN LINE(45,F+10)
      -(238,F+10)
440 NEXT F
450 RETURN
460 '
470 '   linhas verticais
480 '
490 FOR F=33 TO 238 STEP 12
      IF F>33 THEN LINE (F,0)-(F,12)
510   LINE(F,12)-(F,187)
520   IF F>33 THEN LINE (F+1,12)-
      (F+1,187)
530 NEXT F
540 LINE (F-11,0)-(F-11,12)
550 RETURN
560 '
570 '   desenha T
580 '
590 FOR F=45 TO 226 STEP 12
600   FOR G=22 TO 176 STEP 22
610     LINE(F+6,G)-(F+6,G+10)
620     .INF (F+6,G+4)-(F+12,G+4)
630     .INF (F+9,G+4)-(F+9,G+10)
640   NEXT G
650 NEXT F
660 RETURN

```

1000  
 1001  
 1002  
 1003  
 1004  
 1005  
 1006  
 1007  
 1008  
 1009  
 1010  
 1011  
 1012  
 1013  
 1014  
 1015  
 1016  
 1017  
 1018  
 1019  
 1020  
 1021  
 1022  
 1023  
 1024  
 1025  
 1026  
 1027  
 1028  
 1029  
 1030  
 1031  
 1032  
 1033  
 1034  
 1035  
 1036  
 1037  
 1038  
 1039  
 1040  
 1041  
 1042  
 1043  
 1044  
 1045  
 1046  
 1047  
 1048  
 1049  
 1050  
 1051  
 1052  
 1053  
 1054  
 1055  
 1056  
 1057  
 1058  
 1059  
 1060  
 1061  
 1062  
 1063  
 1064  
 1065  
 1066  
 1067  
 1068  
 1069  
 1070  
 1071  
 1072  
 1073  
 1074  
 1075  
 1076  
 1077  
 1078  
 1079  
 1080  
 1081  
 1082  
 1083  
 1084  
 1085  
 1086  
 1087  
 1088  
 1089  
 1090  
 1091  
 1092  
 1093  
 1094  
 1095  
 1096  
 1097  
 1098  
 1099  
 1100  
 1101  
 1102  
 1103  
 1104  
 1105  
 1106  
 1107  
 1108  
 1109  
 1110  
 1111  
 1112  
 1113  
 1114  
 1115  
 1116  
 1117  
 1118  
 1119  
 1120  
 1121  
 1122  
 1123  
 1124  
 1125  
 1126  
 1127  
 1128  
 1129  
 1130  
 1131  
 1132  
 1133  
 1134  
 1135  
 1136  
 1137  
 1138  
 1139  
 1140  
 1141  
 1142  
 1143  
 1144  
 1145  
 1146  
 1147  
 1148  
 1149  
 1150  
 1151  
 1152  
 1153  
 1154  
 1155  
 1156  
 1157  
 1158  
 1159  
 1160  
 1161  
 1162  
 1163  
 1164  
 1165  
 1166  
 1167  
 1168  
 1169  
 1170  
 1171  
 1172  
 1173  
 1174  
 1175  
 1176  
 1177  
 1178  
 1179  
 1180  
 1181  
 1182  
 1183  
 1184  
 1185  
 1186  
 1187  
 1188  
 1189  
 1190  
 1191  
 1192  
 1193  
 1194  
 1195  
 1196  
 1197  
 1198  
 1199  
 1200  
 1201  
 1202  
 1203  
 1204  
 1205  
 1206  
 1207  
 1208  
 1209  
 1210  
 1211  
 1212  
 1213  
 1214  
 1215  
 1216  
 1217  
 1218  
 1219  
 1220  
 1221  
 1222  
 1223  
 1224  
 1225  
 1226  
 1227  
 1228  
 1229  
 1230  
 1231  
 1232  
 1233  
 1234  
 1235  
 1236  
 1237  
 1238  
 1239  
 1240  
 1241  
 1242  
 1243  
 1244  
 1245  
 1246  
 1247  
 1248  
 1249  
 1250  
 1251  
 1252  
 1253  
 1254  
 1255  
 1256  
 1257  
 1258  
 1259  
 1260  
 1261  
 1262  
 1263  
 1264  
 1265  
 1266  
 1267  
 1268  
 1269  
 1270  
 1271  
 1272  
 1273  
 1274  
 1275  
 1276  
 1277  
 1278  
 1279  
 1280  
 1281  
 1282  
 1283  
 1284  
 1285  
 1286  
 1287  
 1288  
 1289  
 1290  
 1291  
 1292  
 1293  
 1294  
 1295  
 1296  
 1297  
 1298  
 1299  
 1300  
 1301  
 1302  
 1303  
 1304  
 1305  
 1306  
 1307  
 1308  
 1309  
 1310  
 1311  
 1312  
 1313  
 1314  
 1315  
 1316  
 1317  
 1318  
 1319  
 1320  
 1321  
 1322  
 1323  
 1324  
 1325  
 1326  
 1327  
 1328  
 1329  
 1330  
 1331  
 1332  
 1333  
 1334  
 1335  
 1336  
 1337  
 1338  
 1339  
 1340  
 1341  
 1342  
 1343  
 1344  
 1345  
 1346  
 1347  
 1348  
 1349  
 1350  
 1351  
 1352  
 1353  
 1354  
 1355  
 1356  
 1357  
 1358  
 1359  
 1360  
 1361  
 1362  
 1363  
 1364  
 1365  
 1366  
 1367  
 1368  
 1369  
 1370  
 1371  
 1372  
 1373  
 1374  
 1375  
 1376  
 1377  
 1378  
 1379  
 1380  
 1381  
 1382  
 1383  
 1384  
 1385  
 1386  
 1387  
 1388  
 1389  
 1390  
 1391  
 1392  
 1393  
 1394  
 1395  
 1396  
 1397  
 1398  
 1399  
 1400  
 1401  
 1402  
 1403  
 1404  
 1405  
 1406  
 1407  
 1408  
 1409  
 1410  
 1411  
 1412  
 1413  
 1414  
 1415  
 1416  
 1417  
 1418  
 1419  
 1420  
 1421  
 1422  
 1423  
 1424  
 1425  
 1426  
 1427  
 1428  
 1429  
 1430  
 1431  
 1432  
 1433  
 1434  
 1435  
 1436  
 1437  
 1438  
 1439  
 1440  
 1441  
 1442  
 1443  
 1444  
 1445  
 1446  
 1447  
 1448  
 1449  
 1450  
 1451  
 1452  
 1453  
 1454  
 1455  
 1456  
 1457  
 1458  
 1459  
 1460  
 1461  
 1462  
 1463  
 1464  
 1465  
 1466  
 1467  
 1468  
 1469  
 1470  
 1471  
 1472  
 1473  
 1474  
 1475  
 1476  
 1477  
 1478  
 1479  
 1480  
 1481  
 1482  
 1483  
 1484  
 1485  
 1486  
 1487  
 1488  
 1489  
 1490  
 1491  
 1492  
 1493  
 1494  
 1495  
 1496  
 1497  
 1498  
 1499  
 1500  
 1501  
 1502  
 1503  
 1504  
 1505  
 1506  
 1507  
 1508  
 1509  
 1510  
 1511  
 1512  
 1513  
 1514  
 1515  
 1516  
 1517  
 1518  
 1519  
 1520  
 1521  
 1522  
 1523  
 1524  
 1525  
 1526  
 1527  
 1528  
 1529  
 1530  
 1531  
 1532  
 1533  
 1534  
 1535  
 1536  
 1537  
 1538  
 1539  
 1540  
 1541  
 1542  
 1543  
 1544  
 1545  
 1546  
 1547  
 1548  
 1549  
 1550  
 1551  
 1552  
 1553  
 1554  
 1555  
 1556  
 1557  
 1558  
 1559  
 1560  
 1561  
 1562  
 1563  
 1564  
 1565  
 1566  
 1567  
 1568  
 1569  
 1570  
 1571  
 1572  
 1573  
 1574  
 1575  
 1576  
 1577  
 1578  
 1579  
 1580  
 1581  
 1582  
 1583  
 1584  
 1585  
 1586  
 1587  
 1588  
 1589  
 1590  
 1591  
 1592  
 1593  
 1594  
 1595  
 1596  
 1597  
 1598  
 1599  
 1600  
 1601  
 1602  
 1603  
 1604  
 1605  
 1606  
 1607  
 1608  
 1609  
 1610  
 1611  
 1612  
 1613  
 1614  
 1615  
 1616  
 1617  
 1618  
 1619  
 1620  
 1621  
 1622  
 1623  
 1624  
 1625  
 1626  
 1627  
 1628  
 1629  
 1630  
 1631  
 1632  
 1633  
 1634  
 1635  
 1636  
 1637  
 1638  
 1639  
 1640  
 1641  
 1642  
 1643  
 1644  
 1645  
 1646  
 1647  
 1648  
 1649  
 1650  
 1651  
 1652  
 1653  
 1654  
 1655  
 1656  
 1657  
 1658  
 1659  
 1660  
 1661  
 1662  
 1663  
 1664  
 1665  
 1666  
 1667  
 1668  
 1669  
 1670  
 1671  
 1672  
 1673  
 1674  
 1675  
 1676  
 1677  
 1678  
 1679  
 1680  
 1681  
 1682  
 1683  
 1684  
 1685  
 1686  
 1687  
 1688  
 1689  
 1690  
 1691  
 1692  
 1693  
 1694  
 1695  
 1696  
 1697  
 1698  
 1699  
 1700  
 1701  
 1702  
 1703  
 1704  
 1705  
 1706  
 1707  
 1708  
 1709  
 1710  
 1711  
 1712  
 1713  
 1714  
 1715  
 1716  
 1717  
 1718  
 1719  
 1720  
 1721  
 1722  
 1723  
 1724  
 1725  
 1726  
 1727  
 1728  
 1729  
 1730  
 1731  
 1732  
 1733  
 1734  
 1735  
 1736  
 1737  
 1738  
 1739  
 1740  
 1741  
 1742  
 1743  
 1744  
 1745  
 1746  
 1747  
 1748  
 1749  
 1750  
 1751  
 1752  
 1753  
 1754  
 1755  
 1756  
 1757  
 1758  
 1759  
 1760  
 1761  
 1762  
 1763  
 1764  
 1765  
 1766  
 1767  
 1768  
 1769  
 1770  
 1771  
 1772  
 1773  
 1774  
 1775  
 1776  
 1777  
 1778  
 1779  
 1780  
 1781  
 1782  
 1783  
 1784  
 1785  
 1786  
 1787  
 1788  
 1789  
 1790  
 1791  
 1792  
 1793  
 1794  
 1795  
 1796  
 1797  
 1798  
 1799  
 1800  
 1801  
 1802  
 1803  
 1804  
 1805  
 1806  
 1807  
 1808  
 1809  
 1810  
 1811  
 1812  
 1813  
 1814  
 1815  
 1816  
 1817  
 1818  
 1819  
 1820  
 1821  
 1822  
 1823  
 1824  
 1825  
 1826  
 1827  
 1828  
 1829  
 1830  
 1831  
 1832  
 1833  
 1834  
 1835  
 1836  
 1837  
 1838  
 1839  
 1840  
 1841  
 1842  
 1843  
 1844  
 1845  
 1846  
 1847  
 1848  
 1849  
 1850  
 1851  
 1852  
 1853  
 1854  
 1855  
 1856  
 1857  
 1858  
 1859  
 1860  
 1861  
 1862  
 1863  
 1864  
 1865  
 1866  
 1867  
 1868  
 1869  
 1870  
 1871  
 1872  
 1873  
 1874  
 1875  
 1876  
 1877  
 1878  
 1879  
 1880  
 1881  
 1882  
 1883  
 1884  
 1885  
 1886  
 1887  
 1888  
 1889  
 1890  
 1891  
 1892  
 1893  
 1894  
 1895  
 1896  
 1897  
 1898  
 1899  
 1900  
 1901  
 1902  
 1903  
 1904  
 1905  
 1906  
 1907  
 1908  
 1909  
 1910  
 1911  
 1912  
 1913  
 1914  
 1915  
 1916  
 1917  
 1918  
 1919  
 1920  
 1921  
 1922  
 1923  
 1924  
 1925  
 1926  
 1927  
 1928  
 1929  
 1930  
 1931  
 1932  
 1933  
 1934  
 1935  
 1936  
 1937  
 1938  
 1939  
 1940  
 1941  
 1942  
 1943  
 1944  
 1945  
 1946  
 1947  
 1948  
 1949  
 1950  
 1951  
 1952  
 1953  
 1954  
 1955  
 1956  
 1957  
 1958  
 1959  
 1960  
 1961  
 1962  
 1963  
 1964  
 1965  
 1966  
 1967  
 1968  
 1969  
 1970  
 1971  
 1972  
 1973  
 1974  
 1975  
 1976  
 1977  
 1978  
 1979  
 1980  
 1981  
 1982  
 1983  
 1984  
 1985  
 1986  
 1987  
 1988  
 1989  
 1990  
 1991  
 1992  
 1993  
 1994  
 1995  
 1996  
 1997  
 1998  
 1999  
 2000  
 2001  
 2002  
 2003  
 2004  
 2005  
 2006  
 2007  
 2008  
 2009  
 2010  
 2011  
 2012  
 2013  
 2014  
 2015  
 2016  
 2017  
 2018  
 2019  
 2020  
 2021  
 2022  
 2023  
 2024  
 2025  
 2026  
 2027  
 2028  
 2029  
 2030  
 2031  
 2032  
 2033  
 2034  
 2035  
 2036  
 2037  
 2038  
 2039  
 2040  
 2041  
 2042  
 2043  
 2044  
 2045  
 2046  
 2047  
 2048  
 2049  
 2050  
 2051  
 2052  
 2053  
 2054  
 2055  
 2056  
 2057  
 2058  
 2059  
 2060  
 2061  
 2062  
 2063  
 2064  
 2065  
 2066  
 2067  
 2068  
 2069  
 2070  
 2071  
 2072  
 2073  
 2074  
 2075  
 2076  
 2077  
 2078  
 2079  
 2080  
 2081  
 2082  
 2083  
 2084  
 2085  
 2086  
 2087  
 2088  
 2089  
 2090  
 2091  
 2092  
 2093  
 2094  
 2095  
 2096  
 2097  
 2098  
 2099  
 2100  
 2101  
 2102  
 2103  
 2104  
 2105  
 2106  
 2107  
 2108  
 2109  
 2110  
 2111  
 2112  
 2113  
 2114  
 2115  
 2116  
 2117  
 2118  
 2119  
 2120  
 2121  
 2122  
 2123  
 2124  
 2125  
 2126  
 2127  
 2128  
 2129  
 2130  
 2131  
 2132  
 2133  
 2134  
 2135  
 2136  
 2137  
 2138  
 2139  
 2140  
 2141  
 2142  
 2143  
 2144  
 2145  
 2146  
 2147  
 2148  
 2149  
 2150  
 2151  
 2152  
 2153  
 2154  
 2155  
 2156  
 2157  
 2158  
 2159  
 2160  
 2161  
 2162  
 2163  
 2164  
 2165  
 2166  
 2167  
 2168  
 2169  
 2170  
 2171  
 2172  
 2173  
 2174  
 2175  
 2176  
 2177  
 2178  
 2179  
 2180  
 2181  
 2182  
 2183  
 2184  
 2185  
 2186  
 2187  
 2188  
 2189  
 2190  
 2191  
 2192  
 2193  
 2194  
 2195  
 2196  
 2197  
 2198  
 2199  
 2200  
 2201  
 2202  
 2203  
 2204  
 2205  
 2206  
 2207  
 2208  
 2209  
 2210  
 2211  
 2212  
 2213  
 2214  
 2215  
 2216  
 2217  
 2218  
 2219  
 2220  
 2221  
 2222  
 2223  
 2224  
 2225  
 2226  
 2227  
 2228  
 2229  
 2230  
 2231  
 2232  
 2233  
 2234  
 2235  
 2236  
 2237  
 2238  
 2239  
 2240  
 2241  
 2242  
 2243  
 2244  
 2245  
 2246  
 2247  
 2248  
 2249  
 2250  
 2251  
 2252  
 2253  
 2254  
 2255  
 2256  
 2257  
 2258  
 2259  
 2260  
 2261  
 2262  
 2263  
 2264  
 2265  
 2266  
 2267  
 2268  
 2269  
 2270  
 2271  
 2272  
 2273  
 2274  
 2275  
 2276  
 2277  
 2278  
 2279  
 2280  
 2281  
 2282  
 2283  
 2284  
 2285  
 2286  
 2287  
 2288  
 2289  
 2290  
 2291  
 2292  
 2293  
 2294  
 2295  
 2296  
 2297  
 2298  
 2299  
 2300  
 2301  
 2302  
 2303  
 2304  
 2305  
 2306  
 2307  
 2308  
 2309  
 2310  
 2311  
 2312  
 2313  
 2314  
 2315  
 2316  
 2317  
 2318  
 2319  
 2320  
 2321  
 2322  
 2323  
 2324  
 2325  
 2326  
 2327  
 2328  
 2329  
 2330  
 2331  
 2332

## 7.7 - SORTEIO DE PALAVRAS DA ROM

O programa apresentado a seguir sorteia uma palavra reservada do BASIC MSX a partir da digitação de alguma tecla pelo usuário. Você pode usá-la em outros programas para obter palavras ao acaso.

100 ' rotina de sorteio	0000
110 A\$=INPUT\$(1) : BEEP	0001
120 A=655361*RND(TIME)	0002
130 A%=163*RND(A)+1	0003
140 EN=14962 : A\$ "" : I=65 : C=0	0004
150 A\$ A\$+CHR\$(I)	0005
160 P=PEEK(EN) : Q=PEEK(EN+1)	0006
170 P\$ CHR\$(P)	0007
180 IF P<128 THEN A\$ A\$+P\$ : GOTO 240	0008
190 A\$ A\$+CHR\$(P-128)	0009
200 EN=EN+1 : C=C+1	0010
210 IF C=A% THEN PRINT A\$ : RUN	0011
220 IF PEEK(EN+1) = 0 THEN 240	0012
230 A\$ "" : A\$ A\$+CHR\$(I)	0013
240 IF PEEK(EN)(>0 THEN 280	0014
250 A\$ "" : I=I+1 : Q\$=CHR\$(I)	0015
260 IF Q\$="J" OR Q\$="Q" THEN 280	0016
270 A\$=A\$+Q\$	0017
280 EN=EN+1	0018
290 IF EN<=15649 THEN 160	0019

ROMA - 1980

## BIBLIOGRAFIA RECOMENDADA,

Coleção de Programas para MSX v 2 - páginas 50 a 55

## 7.8 - SORTEIO DE PALAVRAS EM LINHAS "DATA'S"

O programa apresentado abaixo sorteia uma palavra entre uma coleção de as inseridas em linhas DATA. Experimente acrescentar ao programa outras linhas DATA's com mais palavras. Note que é essencial que a última palavra seja "FIM" para que o programa pare de tentar ler mais palavras.

100	' rotina de sorteio	1104
110	AS INPUT\$(1) : BEEP : RESTORE	1105
120	A=65536!*RND( TIME) : FZ 0	1106
130	READ AS : IF AS="FIM" THEN 150	1107
140	FZ=FZ+1 : GOTO 130	1108
150	AZ=FZ*RND(A)+1 : RESTORE	1109
160	FOR FZ=1 TO AZ	1110
170	READ AS	1111
180	NEXT FZ	1112
190	PRINT AS	1113
200	GOTO 110	1114
1000	REM	1115
1010	DATA BABA,CACA,DADA,FAFA,GAGA,HAHA	1116
1011	DATA JAJA,KAKA,LALA,MAMA,NANA,PAPA	1117
1012	DATA PAPA,RARA,SAGA,TATA,UAVA,VAVA	1118
1013	DATA XAXA,ZAZA,NICK,GUTT,MINHOLETA	1119
1020	DATA FIM	1120

TOTAL = 1120



## 7.9 - MAPEANDO LINHAS REM

O programa apresentado a seguir vasculha a memória RAM do micro a partir de &H8000 à procura de pseudo-instruções REM. Ao encontrar no programa um REM (ou '), a linha em que ele se encontra terá seu número apresentado na tela.

Após digitá-lo, grave-o no formato ASCII e com a numeração bem a ta (como usamos a seguir). Para usá-lo, faça um MERGE com o programa que estiver na memória a comando:

RUN 65100

65100	REM		0000
65110	REM PROCURA LINHAS REM		0001
65120	REM		0002
65130	PRINT CHR\$(12)		0003
65140	EI = 32769		0004
65150	B1 = PEEK (EI)		0005
65160	B2 = PEEK (EI + 1)		0006
65170	B3 = PEEK (EI + 2)		0007
65180	B4 = PEEK (EI + 3)		0008
65190	PL = B1 + 256*B2		0009
65200	NL = B3 + 256*B4		0010
65210	IF B1=0 AND B2=0 THEN 350		0011
65220	PRINT"=====		0012
65230	PRINT" NUL :"; NL		0013
65240	PRINT" CPL :"; PL - EI		0014
65250	PRINT" EPL :"; "&H";HEX\$(PL)		0015
65260	FOR F = (EI + 4) TO (PL - 2)		0016
65270	IF PEEK(F)(>&HBF THEN 310		0017
65280	PRINT,"&H";HEX\$(F); " => REM";		0018
65290	PRINT CHR\$(7)		0019
65300	F = PL - 2		0020
65310	NEXT F		0021
65320	PRINT"-----		0022
65330	EI=PL		0023
65340	GOTO 150		0024
65350	END		0025

00000000 = 00000000

## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX capítulo 1  
Programação Avançada em MSX capítulo 1.

## 7.A - MAPEANDO VARIÁVEIS

O programa apresentado a seguir vasculha a memória RAM do micro a partir da &H8000 a procura de uma dada variável. Se ela existir no programa, a linha em que ela se encontra terá seu número apresentado na tela.

Após digitar-lo, grave-o no formato ASCII e com a numeração bem alta (como usamos a seguir). Para usá-lo faça um MERGE com o programa que estiver na memória e comande:

RUN 65000

65000	REM	
65001	REM MAPA DE VARIÁVEIS	
65002	REM	
65003	PRINT CHR\$(12)	
65004	POKE &HFCAB,1	
65005	INPUT"VARIÁVEL":V\$	
65006	EI = 32769	
65007	B1 = PEEK(EI) : B2 = PEEK(EI + 1)	
65008	B3 = PEEK(EI+2) : B4 = PEEK(EI + 3)	
65009	PL = B1 + 256*B2 : NL = B3 + 256*B4	
65010	IF B1=0 AND B2=0 THEN 65024	
65011	IF PEEK(EI+4)=&H8F THEN 65022	
65012	IF PEEK(EI+5)=&H8F THEN 65022	
65013	L\$="" : IP=0	
65014	FOR F = (EI+4) TO (PL 2)	
65015	IF PEEK(F)=34 THEN IP=(IP+1)	
65016	IF IP/2<>IP\2 THEN 65018	
65017	L\$ = L\$ + CHR\$(PEEK(F))	
65018	NEXT F	
65019	L=INSTR(L\$,V\$)	
65020	IF L=0 THEN 65022	
65021	PRINT ">>>STR\$(NL)+>>> (<<,"	
65022	EI=PL	
65023	GOTO 65007	
65024	POKE &HFCAB,0	
65025	END	

## BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX - capítulo 9.  
Programação Avançada em MSX - capítulo 1.

## 7.8 - MAPEANDO LINHAS DE UM PROGRAMA

O pequeno programa listado a seguir mapeia as linhas de um programa em BASIC presente na memória do micro. Ele deve ser gravado em ASCII para poder ser emendado (com o MERGE) ao final dos programas que se deseja mapear.

Para usá-lo após ter feito o MERGE basta comandar:

```
RUN 60000
```

Os dados serão apresentados em 4 colunas. A 1ª conterá o nº da linha, a 2ª conterá seu endereço inicial, a 3ª seu endereço final e a 4ª o comprimento da linha em bytes.

```
60000 RIM MAPEADOR DE LINHAS
60010 EPL=32768
60020 PRINT "LINHA  E.I.      E.F      Comp."
"
60030 PRINT "-----"
"
60040 NI=PEEK(EPL+2)+256*PEEK(EPL+3)
60050 PRINT NI;">";HEX$(EPL);"a";
60060 C=EPL+1:EPL=PEEK(EPL)+256*PEEK(C)
60070 PRINT HEX$(EPL-1);">";":LPL C+1
60080 IF PEEK(EPL)=0 THEN END
60090 GOTO 60040
```

60000 = 7000

### BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulo 1  
Programação Avançada em MSX - capítulo 1

## 7.C - GERANDO LINHAS DATA

Muitas vezes nos deparamos com situações em que necessitamos armazenar os bytes de uma região da memória em linhas DATA. O programa apresentado a seguir gera na tela, na impressora em disquete ou em cassete, as linhas DATA correspondentes a uma certa região da memória.

Ao ser executado o programa pede a introdução do número da primeira linha DATA a ser gerada e os endereços inicial e final da área da memória a ser lida.

Na forma como está o programa a seguir a listagem das linhas DATA é apresentada na tela. Para fazer com que ela seja enviada à impressora no drive ou ao cassete, deve-se alterar a linha 130, substituindo o nome "CRT," pelo do dispositivo desejado.

```
100 REM GERADATA2
110 REM
120 CLEAR 1000
130 OPEN"CRT:DATAS" FOR OUTPUT AS #1
140 DEFBNQ A-Z
150 L$ = ""
160 INPUT"Entre a linha inicial";L
170 IF L<0 OR L>65000 THEN 160
180 INPUT"Entre o endereço inicial";IS
190 I = VAL("&H"+IS)
200 IF I<&H8000 OR I>&HFFFF THEN 190
210 INPUT"Entre o endereço final";FS
220 F = VAL("&H"+FS)
230 IF F<I OR F>&HFFFF THEN 220
240 L$ = L$ + STR$(L)+" DATA "
250 FOR N=I TO F
260   X$ = "00" + HEX$(PEEK(N))
270   X$ = RIGHT$(X$,2)
280   X = X + 1
290   L$ = L$ + X$
300   IF X/8<>X\8 THEN 350
310     PRINT #1,L$
320     L = L + 10
330     L$ = STR$(L)+" DATA "
340     GOTO 360
350   L$ = L$ + ", "
360 NEXT N
370 IF RIGHT$(L$,1)<>"," THEN END
380 L$ = LEFT$(L$,LEN(L$)-1)
390 PRINT #1,L$
```

100  
110  
120  
130  
140  
150  
160  
170  
180  
190  
200  
210  
220  
230  
240  
250  
260  
270  
280  
290  
300  
310  
320  
330  
340  
350  
360  
370  
380  
390

## 7.D - TROCANDO TOKENS NUM PROGRAMA

Enfrentar o problema de transformar todos os comandos de um programa em outro comando não é uma tarefa das mais agradáveis, principalmente quando o programa é grande.

A rotina a seguir coloca um programa em Linguagem de Máquina a partir do endereço &HC000. Quando executada ela varre a área do programa em BASIC trocando os códigos reais ao conteúdo no endereço &HC09C pelo código contido em &HC09B.

Você pode usar o programa abaixo como uma rotina em BASIC, usando-o com um "MERGE" ou gravando em binário e "pokear" os códigos da tokens manualmente.

Os códigos de todas as tokens pode ser encontrados no livro "APROFUNDANDO-SE NO MSX" nas páginas 67 a 71 ou usando o programa da figura 1.4 do livro "PROGRAMAÇÃO AVANÇADA EM MSX" nas páginas 11 e 12.

Lembre-se que as tokens de funções tem sempre o bit 7 setado antes de serem inseridas num programa; portanto, se você desejar substituir tokens de funções deverá antes acrescentar &H80 ao seu código. Por exemplo para substituir a token da função seno (&H09) pela da função cosseno (&H0C) você deverá "pokear" os valores &H89 e &H8C respectivamente nos endereços &HC09C e &HC09B.

```
50000 '*****
50010 'M BY THE DOCTOR LUZ M
50020 '*****
50030 FOR L &HC000 TO &H109F
50040 READ A$:POKE L,VAL("&H"+A$)
50050 NEXT
50060 DEF USR=&HC000
50070 PRINT"RODAR PROGRAMA ?"
50080 A$=INPUT$(1)
50090 IF A$="Y" THEN GOTO 50100 ELSE END
50100 INPUT"CODIGO DA TOKEN";A
50110 INPUT"NOVO CODIGO";B
50120 POKE &HC09C,A
50130 POKE &HC09B,B
50140 A=USR(0)
50150 LIST
50160 DATA DD,21,FF,7F,DD,23,21,73
50170 DATA C0,DD,7E,00,06,14,8E,20
50180 DATA 21,23,23,10,F9,DD,7E,00
50190 DATA FE,22,28,39,5F,3A,9C,C0
50200 DATA DD,BE,00,20,DF,3A,9B,C0
```

50000  
50010  
50020  
50030  
50040  
50050  
50060  
50070  
50080  
50090  
50100  
50110  
50120  
50130  
50140  
50150  
50160  
50170  
50180  
50190  
50200



```

50210 DATA DD,77,00,7B,FE,84,28,36
50220 DATA 1B,D2,DD,7E,00,FE,00,20
50230 DATA 0E,DD,7E,01,FE,00,20,07
50240 DATA DD,7E,02,FE,00,28,09,23
50250 DATA 7E,47,DD,23,10,FC,18,84
50260 DATA DD,22,9D,C0,C9,DD,23,DD
50270 DATA 7E,00,FE,16,20,A6,FE,00
50280 DATA 20,F3,DD,2B,10,7E,DD,23
50290 DATA DD,7L,00,FE,00,20,F7,DD
50300 DATA 2B,10,91,00,04,0B,03,0C
50310 DATA 03,0D,03,0E,03,0,02,11
50320 DATA 01,12,01,13,01,14,01,15
50330 DATA 01,16,01,17,01,18,01,19
50340 DATA 01,1A,01,1C,03,1D,04,1F
50350 DATA 0B,FF,02,00,00,00,00,20

```

```

MARE
0059
1088
E011
F948
DE
E21
F8E
1108
2108
3145
474F
5E1C
1251
1647

```

TOTAL = 1088



## BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulo 2.  
 Programação Avançada em MSX - capítulo 1.

### 7.6 - AUMENTANDO A VELOCIDADE DE EXECUÇÃO EM BASIC

Para ilustrar alguns métodos de se aumentar a velocidade de execução de programas em BASIC vamos usar uma pequena rotina que calcula os números primos compreendidos entre 2 e 1000. Rode o programa a seguir e anote o tempo que ele levou para rodar.

```

100 TIME=0
110 DIM A(2000)
130 FOR I=2 TO 1000
140   IF A(I)<>0 THEN GOTO 190
150   PRINT I : P=P+1
160   FOR N=2*I TO 2000 STEP I
170     A(N)=1
180   NEXT N
190 NEXT I
200 PRINT : PRINT "TEMPO:";TIME/60
210 END

```

O MSX trabalha normalmente com números em dupla precisão (14 casas) e faz os cálculos em BCD para reduzir ao máximo os erros de arredondamento. Com isso ele perde mais tempo que outros micros menos inteligentes, como os APPLE's, os TRS-80, etc. Mas podemos fazer com que ele use precisão simples (8 casas), uma vez que as operações envolvidas são apenas a adição e a multiplicação de números inteiros. Isso faz o programa ficar ligeiramente mais rápido. Experimente acrescentar ao programa anterior a linha a seguir, rode-o novamente e anote o tempo de execução.

110 DEFSNB A Z

Em nosso caso específico, sabemos que os dados manipulados pelo programa são todos inteiros. Todos os números primos são inteiros. Sabendo disso, podemos fazer com que o MSX dê mais a precisão de seu cálculos um pouco mais. Altere a linha 110 como mostrada a seguir e rode mais uma vez o programa. Você deverá obter um tempo de execução bem menor.

110 DEFINT A-Z

Isso ainda não é tudo. Existem alguns procedimentos que o macro faz ao executar pela primeira vez uma linha do programa que não são refeitos se ela for executada outras vezes. Altere o programa, deixando-o como mostrado a seguir e rode-o mais uma vez.

```
0 TIME=0:KEYOFF:LOCATE0,0,0
1 DEFINT A Z:POKE&HF3B1,3:DIMA(2000)
2 FORI=2TO100
3 IF A(I)() THEN Z
4 PRINT I:P=P+1
5 FORN=2*ITO2000STEP1
6 A(N)=1:NEXT
7 NEXT
8 PRINT:PRINT"TEMPO:";TIME/60
9 END
```

DOI: 10.1002/for

Além da renumeração, da supressão dos espaços em branco e da compactação de várias instruções por linha, o programa foi otimizado com mais algumas instruções.

A linha 8 além de apagar as teclas de funções apaga definitivamente o cursor da tela.

Na linha 1, a instrução POKE na posição &HF3B1 faz com que a tela tenha apenas 3 linhas! Isso reduz bastante o tempo de 'SCROLL' da tela quando ela já está cheia.

Por fim, as instruções NEXT sem especificação do parâmetro nas linhas 6 e 7 também reduzem um pouco o tempo de execução.

Se você tiver um APPLE um TRS 80 ou mesmo um PC, experimente rodar neles um programa equivalente e compare o tempo gasto com o do MSX. A não ser no caso de um PC com clock de 8 Mhz, o MSX será o mais rápido!

## BIBLIOGRAFIA RECOMENDADA.

Curso de Basic MSX v 1      páginas 37 a 38.  
Aprofundando-se no MSX      capítulo 1  
Programação Avançada em MSX - capítulo 1.

## 7.F - "SEARCH", PESQUISADOR DE STRINGS

O programa apresentado a seguir gera, a partir do endereço &HE000, uma rotina em linguagem de Máquina que pode ser chamada pelo BASIC para vasculhar a memória a procura de uma sequência de caracteres (string). Após gerar a rotina em L M, o programa a grava em fita ou em disco com o nome SEARCH ou "SEARCH.BIN", respectivamente.

Para testar o programa, após ele ter sido executado, comande:

```
PRINT,HEX$(USR0("A L E P H"))
```

Ao fazer isso, a string "A L E P H" será procurada na memória e o endereço em que ela for encontrada será mostrado em hexadecimal na tela.

Após ter salvo o programa em BASIC, você pode apagá-lo da memória que mesmo assim a rotina em L M permanecerá funcionando. Para usá-la a sintaxe é sempre a mesma: basta chamar a função USR0 passando como parâmetro a string a ser procurada

```
10 FOR FX=&HE000 TO &HE14B
20 READ A$: POKE FX,VAL("&H"+A$)
30 NEXT FX : DEFUSR0=&HE000
40 BSAVE"SEARCH.BIN",&HE000,&HE14B
50 END
100 DATA FE,03,C0,D5,11,4D,E0,21
110 DATA 4C,E0,36,00,01,FF,00,ED
120 DATA B0,D1,06,00,1A,4F,C5,13
130 DATA 1A,6F,13,1A,67,11,4C,E0
140 DATA ED,80,21,00,00,01,FF,FF
150 DATA 11,4C,E0,1A,ED,B1,28,01
160 DATA C9,13,1A,B7,28,09,ED,A1
170 DATA E0,28,F6,1B,03,1B,E9,AF
180 DATA C1,ED,42,22,F8,F7,3E,02
190 DATA 32,63,F6,C9,41,42,56,41
200 DATA 42,56,53,00,00,00,00,00
210 DATA 00,00,00,00,00,00,00,00
220 DATA 00,00,00,00,00,00,00,00
230 DATA 00,00,00,00,00,00,00,00
240 DATA 00,00,00,00,00,00,00,00
250 DATA 00,00,00,00,00,00,00,00
260 DATA 00,00,00,00,00,00,00,00
270 DATA 00,00,00,00,00,00,00,00
280 DATA 00,00,00,00,00,00,00,00
290 DATA 00,00,00,00,00,00,00,00
300 DATA 00,00,00,00,00,00,00,00
```

1000  
1010  
1020  
1030  
1040  
1050  
1060  
1070  
1080  
1090  
1100  
1110  
1120  
1130  
1140  
1150  
1160  
1170  
1180  
1190  
1200  
1210  
1220  
1230  
1240  
1250  
1260  
1270  
1280  
1290  
1300  
1310  
1320  
1330  
1340  
1350  
1360  
1370  
1380  
1390  
1400  
1410  
1420  
1430  
1440  
1450  
1460  
1470  
1480  
1490  
1500  
1510  
1520  
1530  
1540  
1550  
1560  
1570  
1580  
1590  
1600  
1610  
1620  
1630  
1640  
1650  
1660  
1670  
1680  
1690  
1700  
1710  
1720  
1730  
1740  
1750  
1760  
1770  
1780  
1790  
1800  
1810  
1820  
1830  
1840  
1850  
1860  
1870  
1880  
1890  
1900  
1910  
1920  
1930  
1940  
1950  
1960  
1970  
1980  
1990  
2000  
2010  
2020  
2030  
2040  
2050  
2060  
2070  
2080  
2090  
2100  
2110  
2120  
2130  
2140  
2150  
2160  
2170  
2180  
2190  
2200  
2210  
2220  
2230  
2240  
2250  
2260  
2270  
2280  
2290  
2300  
2310  
2320  
2330  
2340  
2350  
2360  
2370  
2380  
2390  
2400  
2410  
2420  
2430  
2440  
2450  
2460  
2470  
2480  
2490  
2500  
2510  
2520  
2530  
2540  
2550  
2560  
2570  
2580  
2590  
2600  
2610  
2620  
2630  
2640  
2650  
2660  
2670  
2680  
2690  
2700  
2710  
2720  
2730  
2740  
2750  
2760  
2770  
2780  
2790  
2800  
2810  
2820  
2830  
2840  
2850  
2860  
2870  
2880  
2890  
2900  
2910  
2920  
2930  
2940  
2950  
2960  
2970  
2980  
2990  
3000  
3010  
3020  
3030  
3040  
3050  
3060  
3070  
3080  
3090  
3100  
3110  
3120  
3130  
3140  
3150  
3160  
3170  
3180  
3190  
3200  
3210  
3220  
3230  
3240  
3250  
3260  
3270  
3280  
3290  
3300  
3310  
3320  
3330  
3340  
3350  
3360  
3370  
3380  
3390  
3400  
3410  
3420  
3430  
3440  
3450  
3460  
3470  
3480  
3490  
3500  
3510  
3520  
3530  
3540  
3550  
3560  
3570  
3580  
3590  
3600  
3610  
3620  
3630  
3640  
3650  
3660  
3670  
3680  
3690  
3700  
3710  
3720  
3730  
3740  
3750  
3760  
3770  
3780  
3790  
3800  
3810  
3820  
3830  
3840  
3850  
3860  
3870  
3880  
3890  
3900  
3910  
3920  
3930  
3940  
3950  
3960  
3970  
3980  
3990  
4000  
4010  
4020  
4030  
4040  
4050  
4060  
4070  
4080  
4090  
4100  
4110  
4120  
4130  
4140  
4150  
4160  
4170  
4180  
4190  
4200  
4210  
4220  
4230  
4240  
4250  
4260  
4270  
4280  
4290  
4300  
4310  
4320  
4330  
4340  
4350  
4360  
4370  
4380  
4390  
4400  
4410  
4420  
4430  
4440  
4450  
4460  
4470  
4480  
4490  
4500  
4510  
4520  
4530  
4540  
4550  
4560  
4570  
4580  
4590  
4600  
4610  
4620  
4630  
4640  
4650  
4660  
4670  
4680  
4690  
4700  
4710  
4720  
4730  
4740  
4750  
4760  
4770  
4780  
4790  
4800  
4810  
4820  
4830  
4840  
4850  
4860  
4870  
4880  
4890  
4900  
4910  
4920  
4930  
4940  
4950  
4960  
4970  
4980  
4990  
5000  
5010  
5020  
5030  
5040  
5050  
5060  
5070  
5080  
5090  
5100  
5110  
5120  
5130  
5140  
5150  
5160  
5170  
5180  
5190  
5200  
5210  
5220  
5230  
5240  
5250  
5260  
5270  
5280  
5290  
5300  
5310  
5320  
5330  
5340  
5350  
5360  
5370  
5380  
5390  
5400  
5410  
5420  
5430  
5440  
5450  
5460  
5470  
5480  
5490  
5500  
5510  
5520  
5530  
5540  
5550  
5560  
5570  
5580  
5590  
5600  
5610  
5620  
5630  
5640  
5650  
5660  
5670  
5680  
5690  
5700  
5710  
5720  
5730  
5740  
5750  
5760  
5770  
5780  
5790  
5800  
5810  
5820  
5830  
5840  
5850  
5860  
5870  
5880  
5890  
5900  
5910  
5920  
5930  
5940  
5950  
5960  
5970  
5980  
5990  
6000  
6010  
6020  
6030  
6040  
6050  
6060  
6070  
6080  
6090  
6100  
6110  
6120  
6130  
6140  
6150  
6160  
6170  
6180  
6190  
6200  
6210  
6220  
6230  
6240  
6250  
6260  
6270  
6280  
6290  
6300  
6310  
6320  
6330  
6340  
6350  
6360  
6370  
6380  
6390  
6400  
6410  
6420  
6430  
6440  
6450  
6460  
6470  
6480  
6490  
6500  
6510  
6520  
6530  
6540  
6550  
6560  
6570  
6580  
6590  
6600  
6610  
6620  
6630  
6640  
6650  
6660  
6670  
6680  
6690  
6700  
6710  
6720  
6730  
6740  
6750  
6760  
6770  
6780  
6790  
6800  
6810  
6820  
6830  
6840  
6850  
6860  
6870  
6880  
6890  
6900  
6910  
6920  
6930  
6940  
6950  
6960  
6970  
6980  
6990  
7000  
7010  
7020  
7030  
7040  
7050  
7060  
7070  
7080  
7090  
7100  
7110  
7120  
7130  
7140  
7150  
7160  
7170  
7180  
7190  
7200  
7210  
7220  
7230  
7240  
7250  
7260  
7270  
7280  
7290  
7300  
7310  
7320  
7330  
7340  
7350  
7360  
7370  
7380  
7390  
7400  
7410  
7420  
7430  
7440  
7450  
7460  
7470  
7480  
7490  
7500  
7510  
7520  
7530  
7540  
7550  
7560  
7570  
7580  
7590  
7600  
7610  
7620  
7630  
7640  
7650  
7660  
7670  
7680  
7690  
7700  
7710  
7720  
7730  
7740  
7750  
7760  
7770  
7780  
7790  
7800  
7810  
7820  
7830  
7840  
7850  
7860  
7870  
7880  
7890  
7900  
7910  
7920  
7930  
7940  
7950  
7960  
7970  
7980  
7990  
8000  
8010  
8020  
8030  
8040  
8050  
8060  
8070  
8080  
8090  
8100  
8110  
8120  
8130  
8140  
8150  
8160  
8170  
8180  
8190  
8200  
8210  
8220  
8230  
8240  
8250  
8260  
8270  
8280  
8290  
8300  
8310  
8320  
8330  
8340  
8350  
8360  
8370  
8380  
8390  
8400  
8410  
8420  
8430  
8440  
8450  
8460  
8470  
8480  
8490  
8500  
8510  
8520  
8530  
8540  
8550  
8560  
8570  
8580  
8590  
8600  
8610  
8620  
8630  
8640  
8650  
8660  
8670  
8680  
8690  
8700  
8710  
8720  
8730  
8740  
8750  
8760  
8770  
8780  
8790  
8800  
8810  
8820  
8830  
8840  
8850  
8860  
8870  
8880  
8890  
8900  
8910  
8920  
8930  
8940  
8950  
8960  
8970  
8980  
8990  
9000  
9010  
9020  
9030  
9040  
9050  
9060  
9070  
9080  
9090  
9100  
9110  
9120  
9130  
9140  
9150  
9160  
9170  
9180  
9190  
9200  
9210  
9220  
9230  
9240  
9250  
9260  
9270  
9280  
9290  
9300  
9310  
9320  
9330  
9340  
9350  
9360  
9370  
9380  
9390  
9400  
9410  
9420  
9430  
9440  
9450  
9460  
9470  
9480  
9490  
9500  
9510  
9520  
9530  
9540  
9550  
9560  
9570  
9580  
9590  
9600  
9610  
9620  
9630  
9640  
9650  
9660  
9670  
9680  
9690  
9700  
9710  
9720  
9730  
9740  
9750  
9760  
9770  
9780  
9790  
9800  
9810  
9820  
9830  
9840  
9850  
9860  
9870  
9880  
9890  
9900  
9910  
9920  
9930  
9940  
9950  
9960  
9970  
9980  
9990  
10000  
10010  
10020  
10030  
10040  
10050  
10060  
10070  
10080  
10090  
10100  
10110  
10120  
10130  
10140  
10150  
10160  
10170  
10180  
10190  
10200  
10210  
10220  
10230  
10240  
10250  
10260  
10270  
10280  
10290  
10300  
10310  
10320  
10330  
10340  
10350  
10360  
10370  
10380  
10390  
10400  
10410  
10420  
10430  
10440  
10450  
10460  
10470  
10480  
10490  
10500  
10510  
10520  
10530  
10540  
10550  
10560  
10570  
10580  
10590  
10600  
10610  
10620  
10630  
10640  
10650  
10660  
10670  
10680  
10690  
10700  
10710  
10720  
10730  
10740  
10750  
10760  
10770  
10780  
10790  
10800  
10810  
10820  
10830  
10840  
10850  
10860  
10870  
10880  
10890  
10900  
10910  
10920  
10930  
10940  
10950  
10960  
10970  
10980  
10990  
11000  
11010  
11020  
11030  
11040  
11050  
11060  
11070  
11080  
11090  
11100  
11110  
11120  
11130  
11140  
11150  
11160  
11170  
11180  
11190  
11200  
11210  
11220  
11230  
11240  
11250  
11260  
11270  
11280  
11290  
11300  
11310  
11320  
11330  
11340  
11350  
11360  
11370  
11380  
11390  
11400  
11410  
11420  
11430  
11440  
11450  
11460  
11470  
11480  
11490  
11500  
11510  
11520  
11530  
11540  
11550  
11560  
11570  
11580  
11590  
11600  
11610  
11620  
11630  
11640  
11650  
11660  
11670  
11680  
11690  
11700  
11710  
11720  
11730  
11740  
11750  
11760  
11770  
11780  
11790  
11800  
11810  
11820  
11830  
11840  
11850  
11860  
11870  
11880  
11890  
11900  
11910  
11920  
11930  
11940  
11950  
11960  
11970  
11980  
11990  
12000  
12010  
12020  
12030  
12040  
12050  
12060  
12070  
12080  
12090  
12100  
12110  
12120  
12130  
12140  
12150  
12160  
12170  
12180  
12190  
12200  
12210  
12220  
12230  
12240  
12250  
12260  
12270  
12280  
12290  
12300  
12310  
12320  
12330  
12340  
12350  
12360  
12370  
12380  
12390  
12400  
12410  
12420  
12430  
12440  
12450  
12460  
12470  
12480  
12490  
12500  
12510  
12520  
12530  
12540  
12550  
12560  
12570  
12580  
12590  
12600  
12610  
12620  
12630  
12640  
12650  
12660  
12670  
12680  
12690  
12700  
12710  
12720  
12730  
12740  
12750  
12760  
12770  
12780  
12790  
12800  
12810  
12820  
12830  
12840  
12850  
12860  
12870  
12880  
12890  
12900  
12910  
12920  
12930  
12940  
12950  
12960  
12970  
12980  
12990  
13000  
13010  
13020  
13030  
13040  
13050  
13060  
13070  
13080  
13090  
13100  
13110  
13120  
13130  
13140  
13150  
13160  
13170  
13180  
13190  
13200  
13210  
13220  
13230  
13240  
13250  
13260  
13270  
13280  
13290  
13300  
13310  
13320  
13330  
13340  
13350  
13360  
13370  
13380  
13390  
13400  
13410  
13420  
13430  
13440  
13450  
13460  
13470  
13480  
13490  
13500  
13510  
13520  
13530  
13540  
13550  
13560  
13570  
13580  
13590  
13600  
13610  
13620  
13630  
13640  
13650  
13660  
13670  
13680  
13690  
13700  
13710  
13720  
13730  
13740  
13750  
13760  
13770  
13780  
13790  
13800  
13810  
13820  
13830  
13840  
13850  
13860  
13870  
13880  
13890  
13900  
13910  
13920  
13930  
13940  
13950  
13960  
13970  
13980  
13990  
14000  
14010  
14020  
14030  
14040  
14050  
14060  
14070  
14080  
14090  
14100  
14110  
14120  
14130  
14140  
14150  
14160  
14170  
14180  
14190  
14200  
14210  
14220  
14230  
14240  
14250  
14260  
14270  
14280  
14290  
14300  
14310  
14320  
14330  
14340  
14350  
14360  
14370  
14380  
14390  
14400  
14410  
14420  
14430  
14440  
14450  
14460  
14470  
14480  
14490  
14500  
14510  
14520  
14530  
14540  
14550  
14560  
14570  
14580  
14590  
14600  
14610  
14620  
14630  
14640  
14650  
14660  
14670  
14680  
14690  
14700  
14710  
14720  
14730  
14740  
14750  
14760  
14770  
14780  
14790  
14800  
14810  
14820  
14830  
14840  
14850  
14860  
14870  
14880  
14890  
14900  
14910  
14920  
14930  
14940  
14950  
14960  
14970  
14980  
14990  
15000  
15010  
15020  
15030  
15040  
15050  
15060  
15070  
15080  
15090  
15100  
15110  
15120  
15130  
15140  
15150  
15160  
15170  
15180  
15190  
15200  
15210  
15220  
15230  
15240  
15250  
15260  
15270  
15280  
15290  
15300  
15310  
15320  
15330  
15340  
15350  
15360  
15370  
15380  
15390  
15400  
15410  
15420  
15430  
15440  
15450  
15460  
15470  
15480  
15490  
15500  
15510  
15520  
15530  
15540  
15550  
15560  
15570  
15580  
15590  
15600  
15610  
15620  
15630  
15640  
15650  
15660  
15670  
15680  
15690  
15700  
15710  
15720  
15730  
15740  
15750  
15760  
15770  
15780  
15790  
15800  
15810  
15820  
15830  
15840  
15850  
15860  
15870  
15880  
15890  
15900  
15910  
15920  
15930  
15940  
15950  
15960  
15970  
15980  
15990  
16000  
16010  
16020  
16030  
16040  
16050  
16060  
16070  
16080  
16090  
16100  
16110  
16120  
16130  
16140  
16150  
16160  
16170  
16180  
16190  
16200  
16210  
16220  
16230  
16240  
16250  
16260  
16270  
16280  
16290  
16300  
16310  
16320  
16330  
16340  
16350  
16360  
16370  
16380  
16390  
16400  
16410  
16420  
16430  
16440  
16450  
16460  
16470  
16480  
16490  
16500  
16510  
16520  
16530  
16540  
16550  
16560  
16570  
16580  
16590  
16600  
16610  
16620  
16630  
16640  
16650  
16660  
16670  
16680  
16690  
16700  
16710  
16720  
16730  
16740  
16750  
16760  
16770  
16780  
16790  
16800  
16810  
16820  
16830  
16840  
16850  
16860  
16870  
16880  
16890  
16900  
16910  
16920  
16930  
16940  
16950  
16960  
16970  
16980  
16990  
17000  
17010  
17020  
17030  
17040  
17050  
17060  
17070  
17080  
17090  
17100  
17110  
17120  
17130  
17140  
17150  
17160  
17170  
17180  
17190  
17200  
17210  
17220  
17230  
17240  
17250  
17260  
17270  
17280  
17290  
17300  
17310  
17320  
17330  
17340  
17350  
17360  
17370  
17380  
17390  
17400  
17410  
17420  
17430  
17440  
17450  
17460  
17470  
17480  
17490  
17500  
17510  
17520  
17530  
17540  
17550  
17560  
17570  
17580  
17590  
17600  
17610  
17620  
17630  
17640  
17650  
17660  
17670  
17680  
17690  
17700  
17710  
17720  
17730  
17740  
17750  
17760  
17770  
17780  
17790  
17800  
17810  
17820  
17830  
17840  
17850  
17860  
17870  
17880  
17890  
17900  
17910  
17920  
17930  
17940  
17950  
17960  
17970  
17980  
17990  
18000  
18010  
18020  
18030  
18040  
18050  
18060  
18070  
18080  
18090  
18100  
18110  
18120  
18130  
18140  
18150  
18160  
18170  
18180  
18190  
18200  
18210  
18220  
18230  
18240  
18250  
18260  
18270  
18280  
18290  
18300  
18310  
18320  
18330  
18340  
18350  
18360  
1837

```

310 DATA 00,00,00,00,00,00,00,00
320 DATA 00,00,00,00,00,00,00,00
330 DATA 00,00,00,00,00,00,00,00
340 DATA 00,00,00,00,00,00,00,00
350 DATA 00,00,00,00,00,00,00,00
360 DATA 00,00,00,00,00,00,00,00
370 DATA 00,00,00,00,00,00,00,00
380 DATA 00,00,00,00,00,00,00,00
390 DATA 00,00,00,00,00,00,00,00
400 DATA 00,00,00,00,00,00,00,00
410 DATA 00,00,00,00,00,00,00,00
420 DATA 00,00,00,00,00,00,00,00
430 DATA 00,00,00,00,00,00,00,00
440 DATA 00,00,00,00,00,00,00,00
450 DATA 00,00,00,00,00,00,00,00
460 DATA 00,00,00,00,00,00,00,00
470 DATA 00,00,00,00,00,00,00,00
480 DATA 00,00,00,00,00,00,00,00
490 DATA 00,00,00,00,00,00,00,00
500 DATA 00,00,00,00,00,00,00,00
510 DATA 00,00,00,00,00,00,00,00

```

```

E500
E501
E502
E503
E504
E505
E506
E507
E508
E509
E50A
E50B
E50C
E50D
E50E
E50F
E510
E511
E512
E513
E514
E515
E516
E517
E518
E519
E51A
E51B
E51C
E51D
E51E
E51F

```

Experimente, após ter rodado o programa ao menos uma vez desligar o micro, ligá-lo novamente, carregar o programa SEARCH gravado em binário para a memória do micro e comandar:

```
DEFUSR0 -&HE000 : ? HEX% (USR0("color"))
```

Você deverá obter como resultado o endereço da ROM onde existe essa palavra.

## 7.6 - RECUPERANDO PROGRAMAS APAGADOS COM NEW

O programa apresentado a seguir deve ser digitado e gravado. Ao ser executado ele gera e grava um programa em Linguagem de Máquina capaz de recuperar programas em BASIC apagados da memória do micro com o comando NEW.

Existem alguns raros casos em que a rotina não funciona, porém certamente esses casos constituem menos de 1% das situações reais.

Para testar o programa após tê-lo digitado, gravado e executado, comande NEW.

O programa foi "apagado" da memória. NÃO FAÇA NADA MAIS para evitar perder dados que ainda estão na memória do micro (apesar de ele não saber disso)! A primeira coisa a ser feita nessa situação é carregar o programa em Linguagem de Máquina gravado pelo programa em BASIC que você digitou. Comande:

```
G. LOAD "WENNEW.BIN",R
```

Isso será o suficiente para recuperar seu programa em BASIC.

```
100 FOR F=&HE000 TO &HE032
110   READ A$:POKE F,VAL("&H"+A$)
120 NEXT F
130 BSAVE "WENNEW.BIN",&HE000,&HE032
140 END
150 DATA F3,21,04,80,23,7E,FE,00
160 DATA 20,FA,23,23,7E,FE,80,2B
170 DATA 20,F2,22,01,80,E0,1A,6F
180 DATA 13,1A,67,7E,FE,00,20,F5
190 DATA 23,22,C2,F6,22,C4,F6,22
200 DATA C6,F6,21,01,80,22,76,F6
210 DATA FB,C9,FF,00,00,00,00,00
```

0000  
0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
000A  
000B  
000C  
000D  
000E  
000F  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
001A  
001B  
001C  
001D  
001E  
001F  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
002A  
002B  
002C  
002D  
002E  
002F  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0039  
003A  
003B  
003C  
003D  
003E  
003F  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
004A  
004B  
004C  
004D  
004E  
004F  
0050  
0051  
0052  
0053  
0054  
0055  
0056  
0057  
0058  
0059  
005A  
005B  
005C  
005D  
005E  
005F  
0060  
0061  
0062  
0063  
0064  
0065  
0066  
0067  
0068  
0069  
006A  
006B  
006C  
006D  
006E  
006F  
0070  
0071  
0072  
0073  
0074  
0075  
0076  
0077  
0078  
0079  
007A  
007B  
007C  
007D  
007E  
007F  
0080  
0081  
0082  
0083  
0084  
0085  
0086  
0087  
0088  
0089  
008A  
008B  
008C  
008D  
008E  
008F  
0090  
0091  
0092  
0093  
0094  
0095  
0096  
0097  
0098  
0099  
009A  
009B  
009C  
009D  
009E  
009F  
00A0  
00A1  
00A2  
00A3  
00A4  
00A5  
00A6  
00A7  
00A8  
00A9  
00AA  
00AB  
00AC  
00AD  
00AE  
00AF  
00B0  
00B1  
00B2  
00B3  
00B4  
00B5  
00B6  
00B7  
00B8  
00B9  
00BA  
00BB  
00BC  
00BD  
00BE  
00BF  
00C0  
00C1  
00C2  
00C3  
00C4  
00C5  
00C6  
00C7  
00C8  
00C9  
00CA  
00CB  
00CC  
00CD  
00CE  
00CF  
00D0  
00D1  
00D2  
00D3  
00D4  
00D5  
00D6  
00D7  
00D8  
00D9  
00DA  
00DB  
00DC  
00DD  
00DE  
00DF  
00E0  
00E1  
00E2  
00E3  
00E4  
00E5  
00E6  
00E7  
00E8  
00E9  
00EA  
00EB  
00EC  
00ED  
00EE  
00EF  
00F0  
00F1  
00F2  
00F3  
00F4  
00F5  
00F6  
00F7  
00F8  
00F9  
00FA  
00FB  
00FC  
00FD  
00FE  
00FF

0000 0001 0002 0003 0004 0005 0006 0007 0008 0009 000A 000B 000C 000D 000E 000F 0010 0011 0012 0013 0014 0015 0016 0017 0018 0019 001A 001B 001C 001D 001E 001F 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 002A 002B 002C 002D 002E 002F 0030 0031 0032 0033 0034 0035 0036 0037 0038 0039 003A 003B 003C 003D 003E 003F 0040 0041 0042 0043 0044 0045 0046 0047 0048 0049 004A 004B 004C 004D 004E 004F 0050 0051 0052 0053 0054 0055 0056 0057 0058 0059 005A 005B 005C 005D 005E 005F 0060 0061 0062 0063 0064 0065 0066 0067 0068 0069 006A 006B 006C 006D 006E 006F 0070 0071 0072 0073 0074 0075 0076 0077 0078 0079 007A 007B 007C 007D 007E 007F 0080 0081 0082 0083 0084 0085 0086 0087 0088 0089 008A 008B 008C 008D 008E 008F 0090 0091 0092 0093 0094 0095 0096 0097 0098 0099 009A 009B 009C 009D 009E 009F 00A0 00A1 00A2 00A3 00A4 00A5 00A6 00A7 00A8 00A9 00AA 00AB 00AC 00AD 00AE 00AF 00B0 00B1 00B2 00B3 00B4 00B5 00B6 00B7 00B8 00B9 00BA 00BB 00BC 00BD 00BE 00BF 00C0 00C1 00C2 00C3 00C4 00C5 00C6 00C7 00C8 00C9 00CA 00CB 00CC 00CD 00CE 00CF 00D0 00D1 00D2 00D3 00D4 00D5 00D6 00D7 00D8 00D9 00DA 00DB 00DC 00DD 00DE 00DF 00E0 00E1 00E2 00E3 00E4 00E5 00E6 00E7 00E8 00E9 00EA 00EB 00EC 00ED 00EE 00EF 00F0 00F1 00F2 00F3 00F4 00F5 00F6 00F7 00F8 00F9 00FA 00FB 00FC 00FD 00FE 00FF

## BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX - capítulo 1  
Programação Avançada em MSX - capítulo 1.

## 7 H - REDEFININDO MENSAGENS DE ERROS

O programa apresentado a seguir permite a redefinição das mensagens de erro do BASIC. Na verdade ele acrescenta novas mensagens às já existentes. Apenas a título de exemplo, usamos mensagens do tipo "MENSAGEM 1", "MENSAGEM 2", etc. Entretanto você pode criar suas próprias mensagens desde que sempre as termine pelo caractere "H".

100 REM	173
110 REM ERROS ALTERNATIVOS	178
120 REM	180
130 C=AR 200,&HD000	184
140 FOR F=0 TO 61	189
150 READ A\$: A=VAL("&H"+A\$)	18D
160 POKE &HD000H,A	190
170 NEXT F	191
180 READ A\$	193
190 IF A\$="FIM" THEN 250	197
200 FOR G=1 TO LEN(A\$)	198
210 POKE &HD000+F,ASC(MID\$(A\$,G,1))	19C
220 F=F+1	19D
230 NEXT G	19E
240 GOTO 180	19F
250 POKE &HD000+F,255	1A0
260 DEFUSR=&HD000	1A1
270 S=USR(0)	1A2
280 BSAVE "ERROS",&HD000,&HD000+F	1A3
290 END	1A4
300 REM	1A5
310 REM DADOS	1A6
320 REM	1A7
330 DATA 11,0D,00,21,FD,FE,36,C3	1A8
340 DATA 23,73,23,72,C9,EE,F5,C5	1A9
350 DATA 21,3D,00,7E,23,FE,FF,20	1AA
360 DATA 16,FE,23,20,F6,0D,20,F3	1AB
370 DATA 7E,23,FE,FF,20,09,FE,23	1AC
380 DATA 28,05,CD,A2,00,18,F1,3E	1AD
390 DATA 0D,CD,A2,00,3E,0A,C0,A2	1AE
400 DATA 00,C1,F1,E1,C9,23	1AF
410 DATA "MENSAGEM 1" H"	1B0
420 DATA "MENSAGEM 2" H"	1B1
430 DATA "MENSAGEM 3" H"	1B2
440 DATA "MENSAGEM 4" H"	1B3
450 DATA "MENSAGEM 5" H"	1B4
460 DATA "MENSAGEM 6" H"	1B5
470 DATA "MENSAGEM 7" H"	1B6
480 DATA "MENSAGEM 8" H"	1B7

```

490 DATA "MESSAGE" 9
500 DATA "MESSAGE" A
510 DATA "MESSAGE" B
520 DATA "MESSAGE" C
530 DATA "MESSAGE" D
540 DATA "MESSAGE" E
550 DATA "MESSAGE" F
560 DATA "MESSAGE" G
570 DATA "MESSAGE" H
580 DATA "MESSAGE" I
590 DATA "MESSAGE" J
600 DATA "MESSAGE" K
610 DATA "MESSAGE" L
620 DATA "MESSAGE" M
630 DATA "MESSAGE" N
640 DATA "MESSAGE" O
650 DATA "MESSAGE" P
660 DATA "FIN"

```

[illegible]

4490  
7HE4  
7E24  
8071  
8102  
8081  
9410  
9141  
11764  
8901  
0020  
1315  
1604  
25  
344  
134  
11748  
1113

總的來說， $\text{H}_2\text{O}$  和  $\text{H}_2\text{O}_2$  的反應如下：

**BIBLIOGRAFIA RECOMENDADA:**

Aprofundando-se no MSX - pág nas 71 a 76 e 82.



## 7.1 - SOMA SINTÁTICA

Os erros mais frequentes cometidos durante a transcrição de programas listados para a memória do computador são devidos a digitação incorreta por parte do leitor. Raras as vezes em que o erro é devido a falhas de impressão e mais raras ainda as ocasiões em que o programa está realmente com algum erro lógico.

Mesmo considerando o fato de que todos os programas apresentados neste livro estão em BASIC são, em sua maioria, curtos e de fácil digitação, a probabilidade de falhas durante a digitação é algo considerável. Uma simples vírgula substituída involuntariamente por um ponto pode por todo um programa a perder.

Pensando em como diminuir a ocorrência de erros de digitação desenvolvemos uma pequena rotina em Linguagem de Máquina capaz de checar o programa na memória. Antes de continuarmos, digite o programa a seguir exatamente como o listamos abaixo, sem tirar nem por absolutamente nada e tomando o máximo cuidado para não cometer erro algum. Depois, salve o programa em disco ou em fita cassete.

```
100 REM-----
110 REM SOMA SINTÁTICA
120 REM                                     Rubens Jr.
130 REM-----
140 FOR F=&HE000 TO &HE0B5
150   READ A$ : POKE F,VAL("&H"+A$)
160 NEXT F : DEFUSR0=&HE000
170 BSAVE"SOMSIN.BIN",&HE000,&HE0B5
180 SS = USR0(0) : END
190 REM-----
200 DATA 3E,00,32,84,E0,21,00,00
210 DATA 22,80,E0,2A,76,F6,CD,F9
220 DATA 10,CD,63,E0,7A,B3,CA,48
230 DATA E0,ED,53,82,E0,CD,63,E0
240 DATA CD,68,E0,28,28,CD,7E,E0
250 DATA 23,E5,ED,58,82,E0,18,B7
260 DATA ED,52,E1,20,F0,2A,B0,E0
270 DATA CD,94,E0,3E,0D,CD,A2,00
280 DATA 3E,0A,CD,A2,00,2A,B2,E0
290 DATA C3,0E,E0,21,A3,E0,CD,78
300 DATA 66,2A,80,E0,CD,94,E0,3E
310 DATA 02,32,63,F6,2A,80,E0,22
320 DATA F8,F7,C9,5E,23,56,23,C9
330 DATA E5,EB,CD,12,34,3E,09,CD
340 DATA A2,00,3E,3D,CD,A2,00,3E
```

1000  
1100  
1200  
1300  
1400  
1500  
1600  
1700  
1800  
1900  
2000  
2100  
2200  
2300  
2400  
2500  
2600  
2700  
2800  
2900  
3000  
3100  
3200  
3300  
3400

```

350 DATA 09,CD,A2,00,E1,C9,5E,16
360 DATA 00,3A,B4,E0,3C,32,B4,E0
370 DATA AB,5F,E5,2A,B0,E0,19,22
380 DATA B0,E0,E1,C9,22,F8,F7,3E
390 DATA 02,32,63,F6,CD,22,37,CD
400 DATA 7B,66,C9,0D,0A,0D,0A,54
410 DATA 4F,54,41,4C,09,3D,09,00
420 DATA 00,00,00,00,00,42,41,4F

```

```

3425
318C
2189
1661
5568
837E
8745
2296

```

TOTAL = 9360

Com o programa já gravado, comande RUN. Você deverá obter na tela uma listagem terminada com a mensagem:

TOTAL = 9360

Se isso não ocorreu, carregue o programa SOMA SINTÁTICA (previamente gravado) para a memória do micro e confira-o novamente, pois há alguma coisa errada nele.

Após obter o valor correto para a soma TOTAL do programa, grave-o definitivamente.

Um programa em BASIC é armazenado na memória do micro como uma sequência de bytes. A rotina que gera a SOMA SINTÁTICA vai lendo a memória e para cada linha do programa em BASIC calcula uma soma "ponderada" dos bytes que a constituem. O valor com que cada byte participa na soma de cada linha depende de seu próprio valor e da posição que ele ocupa na linha. Para verificar isso, comande NEW e introduza a linha a seguir:

```
10 PRINT "AMOR"
```

A seguir, comande:

```
(S = USR0(0)
```

Você deverá obter a soma TOTAL = 245

Agora, altere a linha 10 para:

```
10 PRINT "ROMA"
```

Comande mais uma vez:

```
SS = USR0(0)
```

A soma TOTAL se a terou para **289** e isso aconteceu apesar de as duas linhas terem exatamente os mesmos caracteres.

Quando o programa tem várias linhas o valor total de cada uma é adicionado na soma da linha subsequente. Desse modo a soma total de um programa é sempre igual a soma ind cada em sua última linha e quando um erro é cometido numa determinada linha as somas de todas as linhas subsequentes são alteradas.

Note porém, que a rotina não é infalível! Calcule como contra-exemplo as somas TOTAIS das duas linhas abaixo:

```
10 PRINT"ACD"
```

e

```
10 PRINT"ABC"
```

Apesar de as linhas serem diferentes, suas SOMAS são iguais!

Quase todos os demais programas deste livro estão acompanhados de sua soma TOTAL obtida através do programa SOMA S NTÁT CA rodando num MSX EXPERT versão 1.1. Se o seu MSX for de outro tipo a soma TOTAL dos programas poderá resultar diferente da que apresentamos pois alguns caracteres (como o Ç por exemplo, podem ter códigos diferentes. Cuidado, portanto se esse for o seu caso!

Outra situação peculiar ocorre com programas que possuem GOTO ou GOSJB. Se calcularmos a soma total desses programas antes de executá-los e depois de executá-los obteremos valores diferentes pois o interpretador BASIC da ROM do MSX usa uma técnica de otimização que altera as linhas com GOTO ou GOSJB após a primeira passada por elas (veja as páginas 16 e 17 do livro PROGRAMAÇÃO AVANÇADA EM MSX para maiores detalhes sobre essa técnica).

Neste livro as somas TOTAIS apresentadas foram sempre calculadas ANTES de se executar o programa sequer uma única vez!

## 7.J - PSEUDO-RAMDISK

O programa apresentado a seguir permite o uso dos 32 Kbytes da RAM não disponíveis para o BASIC (entre os endereços 0 e &H8000). Após digitá-lo e gravá-lo, certificando-se de que ele esteja correto, rode-o. Com isso uma rotina em Linguagem de Máquina estará pronta para ser usada com outros programas, desde que eles não se sobreponham à área de memória entre &H0000 e &H007B.

A rotina permite que os programas presentes na RAM disponível do micro sejam passados para a RAM oculta e posteriormente recuperados de volta para a RAM disponível.

Para passar da RAM ativa para a RAM oculta deve-se comandar:

```
POKE 0,USR0(0Z)
```

Fazendo isso você pode carregar outro programa na memória e usá-lo normalmente, pois o programa anterior estará "salvo" na RAM oculta.

Para recuperar o programa da RAM oculta para a RAM ativa deve-se comandar:

```
POKE 0,USR0(1Z)
```

Com isso o programa "salvo" estará novamente presente na RAM ativa do micro.

Aos desses recursos podemos também "trocar" o conteúdo da RAM ativa e da RAM oculta. Se temos um programa PROC1 "salvo" na RAM oculta e um outro programa PROC2 presente na RAM ativa, para trocá-los de posições devemos comandar:

```
POKE 0,USR0(2Z)
```

Faça alguns testes com programas curtos para se habituar aos comandos.

```
100 SCREEN 0:WIDTH 39:CLEAR 200,&H0000
110 FOR F=&H0000 TO &H007B
120   READ A$:POKE F,VAL("&H"+A$)
130 NEXT F
140 DEFUSR0=&H0000
150 DATA F3,F5,C5,D5,E5,ED,73,FE
160 DATA F3,FE,02,20,5E,73,23,7E
170 DATA FE,00,20,0B,21,00,80,11
180 DATA 00,00,CD,53,D0,18,43,FE
```

```
27-
8-8
128E
1408
1580
2291
21-8
3024
4004
```

```

190 DATA 01,20,0B,21,00,00,11,00
200 DATA 80,CD,53,D0,1B,34,FE,02
210 DATA 20,39,CD,53,D0,01,FF,5F
220 DATA 11,00,00,21,00,B0,1A,32
230 DATA 75,D0,7E,12,3A,75,D0,77
240 DATA 23,13,0B,78,81,20,EF,0B
250 DATA A8,1B,14,DB,A8,47,CB,3F
260 DATA CB,3F,CB,3F,CB,3F,B0,D3
270 DATA AB,C9,01,FF,5F,ED,B0,E6
280 DATA F0,D3,A8,ED,7B,FE,F3,E1
290 DATA D1,C1,F1,FB,C9,00,00,00

```

```

E200
7750
8880
8E00
F287
9052
A511
B359
C1FE
D344
EF43

```

Você pode também programar as teclas de funções com os comandos de SALVAR RECLPERAR e TROCAR programas na RAM oculta para facilitar a operação do programa em L.M

# BIBLIOGRAFIA RECOMENDADA:

Aprofundando-se no MSX - capítulos 0 e 1.  
 Programação Avançada em MSX - capítulo 1

## 7.2 - ROTACIONANDO CARACTERES

A rotina apresentada a seguir permite a escrita com letras 'deitadas' na tela. Melhor do que tentar entender o que o programa faz, é rodá-lo e observar o efeito.

O programa em BASIC serve para gerar uma rotina em Linguagem de Máquina. Ao ser chamada por uma instrução LSR, o caractere cujo código é passado como parâmetro será "girado" de 90°.

Análise as linhas de 200 a 240 para entender melhor como a rotina deve ser usada.

```
10 REM
20 REM GIRA CARACTERES NA SCREEN 1
30 REM
100 CLEAR 300,&HCFFF
110 SCREEN 1
120 DEFINT A
130 DEFUSR0=&HD000
140 FOR F=&HD000 TO &HD033
150   READ A$;A=VAL("&H"+A$)
160   POKE F,A
170 NEXT F
180 INPUT "MENSAGEM";A$
190 PRINT
200 FOR F=1 TO LEN(A$)
210   A=ASC(MID$(A$,F,1))*8
220   PRINT CHR$(A/8)
230   X=USR0(A)
240 NEXT F
250 GOTO 250
260 REM
270 DATA 23,23,5E,23,56,62,6B,E5
280 DATA 01,08,00,11,2C,00,D5,CD
290 DATA 59,00,D1,E1,06,08,C5,06
300 DATA 0B,D5,1B,CB,26,C8,1F,23
310 DATA 10,F9,E8,CD,4D,00,23,D1
320 DATA C1,10,E8,C9,00,00,00,00
330 DATA 00,00,00,00
```

200  
210  
220  
230  
240  
250  
260  
270  
280  
290  
300  
310  
320  
330  
340  
350  
360  
370  
380  
390  
400  
410  
420  
430  
440  
450  
460  
470  
480  
490  
500  
510  
520  
530  
540  
550  
560  
570  
580  
590  
600  
610  
620  
630  
640  
650  
660  
670  
680  
690  
700  
710  
720  
730  
740  
750  
760  
770  
780  
790  
800  
810  
820  
830  
840  
850  
860  
870  
880  
890  
900  
910  
920  
930  
940  
950  
960  
970  
980  
990  
1000

1000 = 1000

## BIBLIOGRAFIA RECOMENDADA.

Linguagem de Máquina MSX - páginas 140 a 143

## 7.1 - ROTINA DE ENTRADA COM INKEY\$

A rotina apresentada a seguir permite a introdução de dados em programas através da instrução INKEY\$.

As linhas de 100 a 170 simulam um programa qualquer. A única linha realmente necessária é a linha 110, onde uma função string para posicionamento do cursor é definida. A rotina de entrada efetivamente começa na linha 1000.

Ao ser chamada a sub-rotina da linha 1000 deve receber a LINHA (na variável LI) e a COLUNA (na variável CO) em que os dados introduzidos deverão aparecer na tela. Além disso deve-se também fornecer quantos caracteres poderão ser introduzidos, isto é, o TAMANHO DA LINHA (na variável TL) e ser introduzida.

Para entrar dados com essa rotina, dispõe-se das seguintes funções:

RETURN - termina a inserção de dados,

BS ou - Volta uma posição apagando,

CLS - Apaga a linha já introduzida.

Ao retornar da sub-rotina (quando se digita RETURN) o programa traz na variável L\$ a linha introduzida.

Essa rotina pode ser particularmente útil nos programas de gerenciamento de dados, em que ao serem introduzidas as informações de cada campo devem ser apresentadas de forma estética na tela.

```
100 SCREEN 0 : WIDTH 39
110 DEF FNPC$(LI,CO)=CHR$(27)+"Y"+
    CHR$(LI+32)+CHR$(CO+32)
120 INPUT "LINHA:";LI
130 INPUT "COLUNA:";CO
140 INPUT "NO CARACTERES:";TL
150 GOSUB 1000
160 PRINT : PRINT,,,,,,,,,"L$=";L$
170 IF STRIG(0) THEN RUN ELSE 170
1000 '
1010 ' SUB ROTINA PARA ENTRAR LINHA
1020 '
1030 L$ = ""
1040 PRINT FNPC$(LI,CO);" ";
1050 IF INKEY$
1060 IF I$="" THEN 1050
1070 IF I$>"-" AND I$<"(" THEN 1230
1080 C=CO+LEN(L$):PRINT FNPC$(LI,C);
```

0174  
0175

0176  
0177  
0178  
0179  
0180  
0181  
0182  
0183  
0184  
0185  
0186  
0187  
0188  
0189  
0190

1090	IF LEN(L\$)<TL THEN PRINT " "	4584
1100	I=ASC(I\$)	4585
1110	IF I <> 8 THEN 1170 ' BS	4586
1120	IF LEN(L\$)=0 THEN 1140	4587
1130	L\$=LEFT\$(L\$,LEN(L\$)-1)	4588
1140	C = CO : PRINT FNPC\$(LI,C);	4589
1150	PRINT L\$;" "	4590
1160	GOTO 1050	4591
1170	IF I <> 12 THEN 1220 'CLS	4592
1180	L\$=""	4593
1190	PRINT FNPC\$(LI,CO);"-";	4594
1200	PRINT STRING\$(TL-1," ")	4595
1210	GOTO 1050	4596
1220	IF I = 13 THEN RETURN 'RETURN	4597
1230	IF LEN(L\$)<TL THEN L\$=L\$+I\$	4598
1240	IF LEN(L\$)=TL THEN L\$=LEFT\$(L\$,TL-1	4599
	) + I\$	4600
1250	C=CO-1+LEN(L\$)	4601
1260	PRINT FNPC\$(LI,C);	4602
1270	PRINT I\$;	4603
1280	IF LEN(L\$)<TL THEN PRINT " "	4604
1290	GOTO 1050	4605

1090 1100 1110 1120 1130 1140 1150 1160 1170 1180 1190 1200 1210 1220 1230 1240 1250 1260 1270 1280 1290



## 7.M - ROTINA PARA MAIÚSCULAS

O programa apresentado abaixo gera uma pequena rotina em Linguagem de Máquina que passa todas as letras comuns minúsculas de uma string para maiúsculas.

Experimente rodá-lo e introduza algumas sequências de letras minúsculas. Note que a string é passada como parâmetro da função USR.

```
100 CLEAR 300,&H8FFF
110 FOR I = &HC000 TO &HC021
120     READ X$
130     POKE I,VAL("&h"+X$)
140 NEXT I
150 DATA 3A,63,F6,FE,03,C0,2A,F8
160 DATA F7,7E,B7,C8,47,23,5E,23
170 DATA 66,6B,7E,FE,61,38,07,FE
180 DATA 7B,30,03,D6,20,77,23,10
190 DATA F1,C9
200 DEFUSR0=&HC000
210 INPUT X$
220 PRINT USR0(X$)
230 GOTO 210
```

8000  
8001  
8002  
8003  
8004  
8005  
8006  
8007  
8008  
8009  
8010  
8011  
8012  
8013  
8014  
8015  
8016  
8017  
8018  
8019  
8020  
8021  
8022  
8023  
8024  
8025  
8026  
8027  
8028  
8029  
8030  
8031  
8032  
8033  
8034  
8035  
8036  
8037  
8038  
8039  
8040  
8041  
8042  
8043  
8044  
8045  
8046  
8047  
8048  
8049  
8050  
8051  
8052  
8053  
8054  
8055  
8056  
8057  
8058  
8059  
8060  
8061  
8062  
8063  
8064  
8065  
8066  
8067  
8068  
8069  
8070  
8071  
8072  
8073  
8074  
8075  
8076  
8077  
8078  
8079  
8080  
8081  
8082  
8083  
8084  
8085  
8086  
8087  
8088  
8089  
8090  
8091  
8092  
8093  
8094  
8095  
8096  
8097  
8098  
8099  
8100  
8101  
8102  
8103  
8104  
8105  
8106  
8107  
8108  
8109  
8110  
8111  
8112  
8113  
8114  
8115  
8116  
8117  
8118  
8119  
8120  
8121  
8122  
8123  
8124  
8125  
8126  
8127  
8128  
8129  
8130  
8131  
8132  
8133  
8134  
8135  
8136  
8137  
8138  
8139  
8140  
8141  
8142  
8143  
8144  
8145  
8146  
8147  
8148  
8149  
8150  
8151  
8152  
8153  
8154  
8155  
8156  
8157  
8158  
8159  
8160  
8161  
8162  
8163  
8164  
8165  
8166  
8167  
8168  
8169  
8170  
8171  
8172  
8173  
8174  
8175  
8176  
8177  
8178  
8179  
8180  
8181  
8182  
8183  
8184  
8185  
8186  
8187  
8188  
8189  
8190  
8191  
8192  
8193  
8194  
8195  
8196  
8197  
8198  
8199  
8200  
8201  
8202  
8203  
8204  
8205  
8206  
8207  
8208  
8209  
8210  
8211  
8212  
8213  
8214  
8215  
8216  
8217  
8218  
8219  
8220  
8221  
8222  
8223  
8224  
8225  
8226  
8227  
8228  
8229  
8230  
8231  
8232  
8233  
8234  
8235  
8236  
8237  
8238  
8239  
8240  
8241  
8242  
8243  
8244  
8245  
8246  
8247  
8248  
8249  
8250  
8251  
8252  
8253  
8254  
8255  
8256  
8257  
8258  
8259  
8260  
8261  
8262  
8263  
8264  
8265  
8266  
8267  
8268  
8269  
8270  
8271  
8272  
8273  
8274  
8275  
8276  
8277  
8278  
8279  
8280  
8281  
8282  
8283  
8284  
8285  
8286  
8287  
8288  
8289  
8290  
8291  
8292  
8293  
8294  
8295  
8296  
8297  
8298  
8299  
8300  
8301  
8302  
8303  
8304  
8305  
8306  
8307  
8308  
8309  
8310  
8311  
8312  
8313  
8314  
8315  
8316  
8317  
8318  
8319  
8320  
8321  
8322  
8323  
8324  
8325  
8326  
8327  
8328  
8329  
8330  
8331  
8332  
8333  
8334  
8335  
8336  
8337  
8338  
8339  
8340  
8341  
8342  
8343  
8344  
8345  
8346  
8347  
8348  
8349  
8350  
8351  
8352  
8353  
8354  
8355  
8356  
8357  
8358  
8359  
8360  
8361  
8362  
8363  
8364  
8365  
8366  
8367  
8368  
8369  
8370  
8371  
8372  
8373  
8374  
8375  
8376  
8377  
8378  
8379  
8380  
8381  
8382  
8383  
8384  
8385  
8386  
8387  
8388  
8389  
8390  
8391  
8392  
8393  
8394  
8395  
8396  
8397  
8398  
8399  
8400  
8401  
8402  
8403  
8404  
8405  
8406  
8407  
8408  
8409  
8410  
8411  
8412  
8413  
8414  
8415  
8416  
8417  
8418  
8419  
8420  
8421  
8422  
8423  
8424  
8425  
8426  
8427  
8428  
8429  
8430  
8431  
8432  
8433  
8434  
8435  
8436  
8437  
8438  
8439  
8440  
8441  
8442  
8443  
8444  
8445  
8446  
8447  
8448  
8449  
8450  
8451  
8452  
8453  
8454  
8455  
8456  
8457  
8458  
8459  
8460  
8461  
8462  
8463  
8464  
8465  
8466  
8467  
8468  
8469  
8470  
8471  
8472  
8473  
8474  
8475  
8476  
8477  
8478  
8479  
8480  
8481  
8482  
8483  
8484  
8485  
8486  
8487  
8488  
8489  
8490  
8491  
8492  
8493  
8494  
8495  
8496  
8497  
8498  
8499  
8500  
8501  
8502  
8503  
8504  
8505  
8506  
8507  
8508  
8509  
8510  
8511  
8512  
8513  
8514  
8515  
8516  
8517  
8518  
8519  
8520  
8521  
8522  
8523  
8524  
8525  
8526  
8527  
8528  
8529  
8530  
8531  
8532  
8533  
8534  
8535  
8536  
8537  
8538  
8539  
8540  
8541  
8542  
8543  
8544  
8545  
8546  
8547  
8548  
8549  
8550  
8551  
8552  
8553  
8554  
8555  
8556  
8557  
8558  
8559  
8560  
8561  
8562  
8563  
8564  
8565  
8566  
8567  
8568  
8569  
8570  
8571  
8572  
8573  
8574  
8575  
8576  
8577  
8578  
8579  
8580  
8581  
8582  
8583  
8584  
8585  
8586  
8587  
8588  
8589  
8590  
8591  
8592  
8593  
8594  
8595  
8596  
8597  
8598  
8599  
8600  
8601  
8602  
8603  
8604  
8605  
8606  
8607  
8608  
8609  
8610  
8611  
8612  
8613  
8614  
8615  
8616  
8617  
8618  
8619  
8620  
8621  
8622  
8623  
8624  
8625  
8626  
8627  
8628  
8629  
8630  
8631  
8632  
8633  
8634  
8635  
8636  
8637  
8638  
8639  
8640  
8641  
8642  
8643  
8644  
8645  
8646  
8647  
8648  
8649  
8650  
8651  
8652  
8653  
8654  
8655  
8656  
8657  
8658  
8659  
8660  
8661  
8662  
8663  
8664  
8665  
8666  
8667  
8668  
8669  
8670  
8671  
8672  
8673  
8674  
8675  
8676  
8677  
8678  
8679  
8680  
8681  
8682  
8683  
8684  
8685  
8686  
8687  
8688  
8689  
8690  
8691  
8692  
8693  
8694  
8695  
8696  
8697  
8698  
8699  
8700  
8701  
8702  
8703  
8704  
8705  
8706  
8707  
8708  
8709  
8710  
8711  
8712  
8713  
8714  
8715  
8716  
8717  
8718  
8719  
8720  
8721  
8722  
8723  
8724  
8725  
8726  
8727  
8728  
8729  
8730  
8731  
8732  
8733  
8734  
8735  
8736  
8737  
8738  
8739  
8740  
8741  
8742  
8743  
8744  
8745  
8746  
8747  
8748  
8749  
8750  
8751  
8752  
8753  
8754  
8755  
8756  
8757  
8758  
8759  
8760  
8761  
8762  
8763  
8764  
8765  
8766  
8767  
8768  
8769  
8770  
8771  
8772  
8773  
8774  
8775  
8776  
8777  
8778  
8779  
8780  
8781  
8782  
8783  
8784  
8785  
8786  
8787  
8788  
8789  
8790  
8791  
8792  
8793  
8794  
8795  
8796  
8797  
8798  
8799  
8800  
8801  
8802  
8803  
8804  
8805  
8806  
8807  
8808  
8809  
8810  
8811  
8812  
8813  
8814  
8815  
8816  
8817  
8818  
8819  
8820  
8821  
8822  
8823  
8824  
8825  
8826  
8827  
8828  
8829  
8830  
8831  
8832  
8833  
8834  
8835  
8836  
8837  
8838  
8839  
8840  
8841  
8842  
8843  
8844  
8845  
8846  
8847  
8848  
8849  
8850  
8851  
8852  
8853  
8854  
8855  
8856  
8857  
8858  
8859  
8860  
8861  
8862  
8863  
8864  
8865  
8866  
8867  
8868  
8869  
8870  
8871  
8872  
8873  
8874  
8875  
8876  
8877  
8878  
8879  
8880  
8881  
8882  
8883  
8884  
8885  
8886  
8887  
8888  
8889  
8890  
8891  
8892  
8893  
8894  
8895  
8896  
8897  
8898  
8899  
8900  
8901  
8902  
8903  
8904  
8905  
8906  
8907  
8908  
8909  
8910  
8911  
8912  
8913  
8914  
8915  
8916  
8917  
8918  
8919  
8920  
8921  
8922  
8923  
8924  
8925  
8926  
8927  
8928  
8929  
8930  
8931  
8932  
8933  
8934  
8935  
8936  
8937  
8938  
8939  
8940  
8941  
8942  
8943  
8944  
8945  
8946  
8947  
8948  
8949  
8950  
8951  
8952  
8953  
8954  
8955  
8956  
8957  
8958  
8959  
8960  
8961  
8962  
8963  
8964  
8965  
8966  
8967  
8968  
8969  
8970  
8971  
8972  
8973  
8974  
8975  
8976  
8977  
8978  
8979  
8980  
8981  
8982  
8983  
8984  
8985  
8986  
8987  
8988  
8989  
8990  
8991  
8992  
8993  
8994  
8995  
8996  
8997  
8998  
8999  
9000  
9001  
9002  
9003  
9004  
9005  
9006  
9007  
9008  
9009  
9010  
9011  
9012  
9013  
9014  
9015  
9016  
9017  
9018  
9019  
9020  
9021  
9022  
9023  
9024  
9025  
9026  
9027  
9028  
9029  
9030  
9031  
9032  
9033  
9034  
9035  
9036  
9037  
9038  
9039  
9040  
9041  
9042  
9043  
9044  
9045  
9046  
9047  
9048  
9049  
9050  
9051  
9052  
9053  
9054  
9055  
9056  
9057  
9058  
9059  
9060  
9061  
9062  
9063  
9064  
9065  
9066  
9067  
9068  
9069  
9070  
9071  
9072  
9073  
9074  
9075  
9076  
9077  
9078  
9079  
9080  
9081  
9082  
9083  
9084  
9085  
9086  
9087  
9088  
9089  
9090  
9091  
9092  
9093  
9094  
9095  
9096  
9097  
9098  
9099  
9100  
9101  
9102  
9103  
9104  
9105  
9106  
9107  
9108  
9109  
9110  
9111  
9112  
9113  
9114  
9115  
9116  
9117  
9118  
9119  
9120  
9121  
9122  
9123  
9124  
9125  
9126  
9127  
9128  
9129  
9130  
9131  
9132  
9133  
9134  
9135  
9136  
9137  
9138  
9139  
9140  
9141  
9142  
9143  
9144  
9145  
9146  
9147  
9148  
9149  
9150  
9151  
9152  
9153  
9154  
9155  
9156  
9157  
9158  
9159  
9160  
9161  
9162  
9163  
9164  
9165  
9166  
9167  
9168  
9169  
9170  
9171  
9172  
9173  
9174  
9175  
9176  
9177  
9178  
9179  
9180  
9181  
9182  
9183  
9184  
9185  
9186  
9187  
9188  
9189  
9190  
9191  
9192  
9193  
9194  
9195  
9196  
9197  
9198  
9199  
9200  
9201  
9202  
9203  
9204  
9205  
9206  
9207  
9208  
9209  
9210  
9211  
9212  
9213  
9214  
9215  
9216  
9217  
9218  
9219  
9220  
9221  
9222  
9223  
9224  
9225  
9226  
9227  
9228  
9229  
9230  
9231  
9232  
9233  
9234  
9235  
9236  
9237  
9238  
9239  
9240  
9241  
9242  
9243  
9244  
9245  
9246  
9247  
9248  
9249  
9250  
9251  
9252  
9253  
9254  
9255  
9256  
9257  
9258  
9259  
9260  
9261  
9262  
9263  
9264  
9265  
9266  
9267  
9268  
9269  
9270  
9271  
9272  
9273  
9274  
9275  
9276  
9277  
9278  
9279  
9280  
9281  
9282  
9283  
9284  
9285  
9286  
9287  
9288  
9289  
9290  
9291  
9292  
9293  
9294  
9295  
9296  
9297  
9298  
9299  
9300  
9301  
9302  
9303  
9304  
9305  
9306  
9307  
9308  
9309  
9310  
9311  
9312  
9313  
9314  
9315  
9316  
9317  
9318  
9319  
9320  
9321  
9322  
9323  
9324  
9325  
9326  
9327  
9328  
9329  
9330  
9331  
9332  
9333  
9334  
9335  
9336  
9337  
9338  
9339  
9340  
9341  
9342  
9343  
9344  
9345  
9346  
9347  
9348  
9349  
9350  
9351  
9352  
9353  
9354  
9355  
9356  
9357  
9358  
9359  
9360  
9361  
9362  
9363  
9364  
9365  
9366  
9367  
9368  
9369  
9370  
9371  
9372  
9373  
9374  
9375  
9376  
9377  
9378  
9379  
9380  
9381  
9382  
9383  
9384  
9385  
9386  
9387  
9388  
9389  
9390  
9391  
9392  
9393  
9394  
9395  
9396  
9397  
9398  
9399  
9400  
9401  
9402  
9403  
9404  
9405  
9406  
9407  
9408  
9409  
9410  
9411  
9412  
9413  
9414  
9415  
9416  
9417  
9418  
9419  
9420  
9421  
9422  
9423  
9424  
9425  
9426  
9427  
9428  
9429  
9430  
9431  
9432  
9433  
9434  
9435  
9436  
9437  
9438  
9439  
9440  
9441  
9442  
9443  
9444  
9445  
9446  
9447  
9448  
9449  
9450  
9451  
9452  
9453  
9454  
9455  
9456  
9457  
9458  
9459  
9460  
9461  
9462  
9463  
9464  
9465  
9466  
9467  
9468  
9469  
9470  
9471  
9472  
9473  
9474  
9475  
9476  
9477  
9478  
9479  
9480  
9481  
9482  
9483  
9484  
9485  
9486  
9487  
9488  
9489  
9490  
9491  
9492  
9493  
9494  
9495  
9496  
9497  
9498  
9499  
9500  
9501  
9502  
9503  
9504  
9505  
9506  
9507  
9508  
9509  
9510  
9511  
9512  
9513  
9514  
9515  
9516  
9517  
9518  
9519  
9520  
9521  
9522  
9523  
9524  
9525  
9526  
9527  
9528  
9529  
9530  
9531  
9532  
9533  
9534  
9535  
9536  
9537  
9538  
9539  
9540  
9541  
9542  
9543  
9544  
9545  
9546  
9547  
9548  
9549  
9550  
9551  
9552  
9553  
9554  
9555  
9556  
9557  
9558  
9559  
9560  
9561  
9562  
9563  
9564  
9565  
9566  
9567  
9568  
9569  
9570  
9571  
9572  
9573  
9574  
9575  
9576  
9577  
9578  
9579  
9580  
9581  
9582  
9583  
9584  
9585  
9586  
9587  
9588  
9589  
9590  
9591  
9592  
9593  
9594  
9595  
9596  
9597  
9598  
9599  
9600  
9601  
9602  
9603  
9604  
9605  
9606  
9607  
9608  
9609  
9610  
9611  
9612  
9613  
9614  
9615  
9616  
9617  
9618  
9619  
9620  
9621  
9622  
9623  
9624  
9625  
9626  
9627  
9628  
9629  
9630  
9631  
9632  
9633  
9634  
9635  
9636  
9637  
9638  
9639  
9640  
9641  
9642  
9643  
9644  
9645  
9646  
9647  
9648  
9649  
9650  
9651  
9652  
9653  
9654  
9655  
9656  
9657  
9658  
9659  
9660  
9661  
9662  
9663  
9664  
9665  
9666  
9667  
9668  
9669  
9670  
9671  
9672  
9673  
9674  
9675  
9676  
9677  
9678  
9679  
9680  
9681  
9682  
9683  
9684  
9685  
9686  
9687  
9688  
9689  
9690  
9691  
9692  
9693  
9694  
9695  
9696  
9697  
9698  
9699  
9700  
9701  
9702  
9703  
9704  
9705  
9706  
9707  
9708  
9709  
9710  
9711  
9712  
9713  
9714  
9715  
9716  
9717  
9718  
9719  
9720  
9721  
9722  
9723  
9724  
9725  
9726  
9727  
9728  
9729  
9730  
9731  
9732  
9733  
9734  
9735  
9736  
9737  
9738  
9739  
9740  
9741  
9742  
9743  
9744  
9745  
9746  
9747  
9748  
9749  
9750  
9751  
9752  
9753  
9754  
9755  
9756  
9757  
9758  
9759  
9760  
9761  
9762  
9763  
9764  
9765  
9766  
9767  
9768  
9769  
9770  
9771  
9772  
9773  
9774  
9775  
9776  
9777  
9778  
9779  
9780  
9781  
9782  
9783  
9784  
9785  
9786  
9787  
9788  
9789  
9790  
9791  
9792  
9793  
9794  
9795  
9796  
9797  
9798  
9799  
9800  
9801  
9802  
9803  
9804  
9805  
9806  
9807  
9808  
9809  
9810  
9811  
9812  
9813  
9814  
9815  
9816  
9817  
9818  
9819  
9820  
9821  
9822  
9823  
9824  
9825  
9826  
9827  
9828  
9829  
9830  
9831  
9832  
9833  
9834  
9835  
9836  
9837  
9838  
9839  
9840  
9841  
9842  
9843  
9844  
9845  
9846  
9847  
9848  
9849  
9850  
9851  
9852  
9853  
9854  
9855  
9856  
9857  
9858  
9859  
9860  
9861  
9862  
9863  
9864  
9865  
9866  
9867  
9868  
9869  
9870  
9871  
9872  
9873  
9874  
9875  
9876  
9877  
9878  
9879  
9880  
9881  
9882  
9883  
9884  
9885  
9886  
9887  
9888  
9889  
9890  
9891  
9892  
9893  
9894  
9895  
9896  
9897  
9898  
9899  
9900  
9901  
9902  
9903  
9904  
9905  
9906  
9907  
9908  
9909  
9910  
9911  
9912  
9913  
9914  
9915  
9916  
9917  
9918  
9919  
9920  
9921  
9922  
9923  
9924  
9925  
9926  
9927  
9928  
9929  
9930  
9931  
9932  
9933  
9934  
9935  
9936  
9937  
9938  
9939  
9940  
9941  
9942  
9943  
9944  
9945  
9946  
9947  
9948  
9949  
9950  
9951  
9952  
9953  
9954  
9955  
9956  
9957  
9958  
9959  
9960  
9961  
9962  
9963  
9964  
9965  
9966  
9967  
9968  
9969  
9970  
9971  
9972  
9973  
9974  
9975  
9976  
9977  
9978  
9979  
9980  
9981  
9982  
9983  
9984  
9985  
9986  
9987

## 7.4 - REDUZINDO A TELA A UM CARACTERE

Você sabe que podemos usar o comando WIDTH para as telas de texto de modo a fazê-las ficar com apenas uma coluna. Experimente comandar:

```
WIDTH 1
```

Agora a tela tem apenas uma coluna. Para voltar ao normal, basta usar novamente o comando WIDTH.

Um fato menos conhecido é a possibilidade de fazer com que a tela tenha apenas 1 linha. Para isso, entretanto, não existe um comando dedicado do BASIC, e é necessário usar o comando POKE para alterar o valor de uma variável do sistema, a CRTCNT, em &HF3B1.

Experimente comandar:

```
POKE &HF3B1,1
```

Com isso a tela deverá ficar com apenas uma linha.

Para reduzir a tela a apenas um caractere, basta usar o par de comandos:

```
WIDTH1:POKE&HF3B1,1
```

Isso, entretanto, tornará o teclado inoperante e você perderá o controle sobre o micro. Portanto, cuidado ao usar esta dica!



## BIBLIOGRAFIA RECOMENDADA.

Aprofundando-se no MSX      pagina 47

## 7.0 - GRANDES EXPOENTES

Recentemente, astrônomos americanos detectaram o que parecem ser galáxias em formação a 17 bilhões de anos luz da Terra! Estes objetos estariam quase no limite do Universo observável cujo raio hoje é estimado em 20 bilhões de anos-luz.

Aproveitando que você tem um computador na frente você resolve calcular o raio do Universo em metros.

O cálculo é simples (pelo menos no computador!) você sabe que a luz se propaga com uma velocidade de 300 000 quilômetros por segundo. Basta portanto multiplicar esta velocidade de 300 000 000 m/s pe o número de segundos em 20 bilhões de anos (lembre-se que o ano tem 365 dias de 24 horas de 60 minutos de 60 segundos).

O seguinte então, o seguinte programinha:

```
10 R=30000000000#*200000000000#*365*24*60*60
20 PRINT"RAIO DO UNIVERSO=";R;"METROS"
```

Rodando-o você deve obter a coisa daí quant a de 1 89216 vezes 10 elevado à potência 26!

Entusiasmado com a rapidez de cálculo do seu MSX você resolve calcular o volume do Universo (supondo ingenuamente que ele seja esférico)

Lembrando que o volume de uma esfera é dado pela expressão:

$$V = 4/3 \cdot \pi \cdot R^3$$

Você então completa seu programinha com as seguintes linhas:

```
30 PI=4*ATN(1)
40 X=(4/3)*PI
50 V=X*R^3
60 PRINT"VOLUME DO UNIVERSO=";V;"m3"
```

Ao rodar seu programa assim incrementado você tem a decepção de obter um "overflow n 50", pois você "estourou" a capacidade de cálculo do MSX.

Nesta dica, vamos apresentar como contornar o problema de expoentes grandes demais. Basta lembrar

QJ8r

$$\log (A \times B) = \log A + \log B$$

$$\log A^n = n \times \log A$$

Substitua as linhas 50 e 60 por:

```
100 LV=LOG(X)+3*LOG(R)
110 LD LV/LOG(10)
120 E=INT(LD)
130 M=10^(LD-E)
140 A$=STR$(M)+"E"+"+STR$(E)
150 PRINT"VOL. DO UNIVERSO=";A$;"M.C."
```

Na linha 100 você calcula o logaritmo neperiano do volume (o MSX só trabalha com logaritmos naturais). Na linha 110 você transforma o log natural em log decimal (é só dividir por LOG(10); esta é mais uma dica importante).

Para entender as linhas 120 e 130, lembre-se que se, por exemplo,

$$\log x = 79,4529$$

então

$$x = 10^{0,4529} \times 10^{79}$$

Portanto, toda vez que você se deparar com números demasiadamente grandes para seu MSX basta calcular seu logaritmo decimal, pegar a parte inteira com expoente de 10 e descobrir o multiplicando na frente dele operando como aprendemos





Como você deve ter percebido, as dicas deste livro foram apresentadas de forma bem prática e resumida, de modo a poderem ser usadas imediatamente. Se você quiser obter maiores detalhes sobre o funcionamento de cada uma delas é recomendável estudar detalhadamente os livros citados na "BIBLIOGRAFIA RECOMENDADA", ao final de cada dica. A seguir apresentamos um resumo do conteúdo de cada um desses livros para melhor orientá-lo.

### LINGUAGEM BASIC MSX

Uma "enciclopédia" do BASIC MSX, com a sintaxe, função e exemplo de cada palavra do BASIC MSX.

### CURSO DE BASIC MSX v.1

Uma introdução clara e didática ao BASIC residente do MSX, apresentada em 8 aulas com exercícios (e suas respostas!).

### COLEÇÃO DE PROGRAMAS PARA MSX v.1 e v.2

Programas didáticos, aplicativos e utilitários explicados passo a passo para que o leitor aprenda a fazer seus próprios programas.

### APROFUNDANDO-SE NO MSX

O "best seller" da literatura técnica sobre MSX, com a descrição detalhada da arquitetura da máquina e de cada uma de suas partes.

### PROGRAMAÇÃO AVANÇADA EM MSX

Exemplos e rotinas utilitárias em ASSEMBLY ensinando ao leitor como se obtém o máximo das máquinas MSX.

### LINGUAGEM DE MÁQUINA MSX

Uma introdução completa e didática aos poderosos recursos da Linguagem de Máquina Z80 aplicada aos micros MSX. Contém as instruções secretas do Z80.

### USANDO O DISK DRIVE NO MSX

O MSXDOS, O CP/M para MSX e o DISK BASIC comentados exaustivamente.

### SISTEMA DE DISCO PARA MSX

O SOLXDOS e o BASIC de DISCO comentados passo a passo de forma clara e didática.

### DRIVES LEOPARD DE 3 1/2"

O primeiro livro sobre drives de 3 1/2" editado no Brasil. Contém todos os recursos do MSXDOS e do DISK BASIC MSX aplicados aos drives de 3 1/2".

Para receber gratuitamente o boletim informativo da ALEPH, contendo dicas de programação, artigos técnicos e informações sobre os últimos lançamentos para seu micro, envie seu nome e endereço completos (incluindo o CEP) para:

**EDITORA ALEPH**  
**Caixa Postal: 20.707**  
**01498 São Paulo SP**

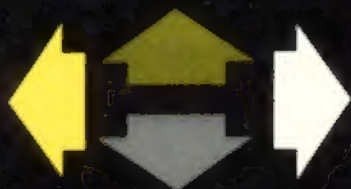
Se você quiser adquirir todos os programas com mais de 512 bytes listados neste livro já gravados em DISCO (apenas em disco!!!), entre em contato conosco.

Para comprar nossos livros pelo correio, informe-se escrevendo ou telefonando para nós.

Nosso telefone é:

**(011) 843-3202**

**COLEÇÃO MSX**



## **100 DICAS PARA MSX**

Ao longo de dois anos trabalhando com micros MSX pudemos atender a milhares (sem exagero!) de dúvidas sobre essas máquinas.

Esquentando a orelha ao telefone, atolando em montanhas de cartas, ou atendendo pessoalmente nossos leitores, surgiram as mais de cem dicas publicadas neste livro, como uma espécie de resposta coletiva.

Apesar de apelarmos frequentemente para o uso da Linguagem de Máquina, todos os programas listados neste livro estão em BASIC, prontos para serem usados e com checagem automática para detecção de erros de digitação.

Esperamos com isso abrir novos horizontes aos nossos leitores, tanto aos mais experientes quanto aos principiantes.

